

digit

FastTrack

YOUR HANDY GUIDE TO EVERYDAY TECHNOLOGY

to

WORDPRESS

Arguably the most popular **blogging tool**, WordPress can be **your platform of choice too**. We show you how

Installing and configuring WordPress
Managing a blog

Options, Comments & Accounts

Themes

Plugins

Widgets

CMS

Dealing with databases

bbPress, BuddyPress & BackPress



OCTOBER 2010						
SUN	MON	TUES	WED	THURS	FRI	SAT
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30



digit

YOUR TECHNOLOGY NAVIGATOR

thinkdigit.com

Fast Track

To



WORD PRESS

CREDITS

The People Behind This Book

EDITORIAL

Editor	Robert Sovereign-Smith
Head-Copy Desk	Nash David
Writers	Hanu Kunduru, Rahil Banthia, Siddarth Raman

DESIGN AND LAYOUT

Lead Designer	Vijay Padaya
Senior Designer	Baiju NV
Cover Design	Suresh Kumar

© 9.9 Mediaworx Pvt. Ltd.

Published by 9.9 Mediaworx

No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means without the prior written permission of the publisher.

October 2010

Free with Digit. Not to be sold separately. If you have paid separately for this book, please email the editor at editor@thinkdigit.com along with details of location of purchase, for appropriate action.

Contents

	INSTALLING & CONFIGURING	07
	MANAGING A BLOG	12
	THEMES	21
	PLUGINS.....	30
	WIDGETS.....	44
	CONTENT MANAGEMENT SYSTEM.....	50
	DEALING WITH DATABASES.....	60
	MANAGING A BLOG	70
	BBPRESS, BUDDYPRESS & BACKPRESS	
ON THE MINDWARE DVD	

INTRODUCTION

Communicating with others beyond the basic needs of survival is one of the basic characteristic that distinguishes humans from others in the realm of *creation*. With the advent of web 2.0 people willing to express themselves found a convenient platform in the form of blogging. It is said never before was there so little to say by so many people in so many words. Jokes apart, blogging is serious business and tools like WordPress make it easier.

Free Open Source blog tools like WordPress help you create and manage your blog site with near zero effort. WordPress is an open source CMS, often used as a blog publishing application powered by PHP and MySQL. It has many features including a plug in architecture and a templating (allows for widgets and themes among many other things) system. WordPress is the most popular blog software in use today.

Talking a little about the history of this beautiful tool, WordPress first appeared in 2003 as a collaboration between Matt Mullenweg and Mike Little in an attempt to folk b2 or cafelog, the precursor of WordPress. The name 'WordPress' was an offering by Christine Selleck, a friend of Mullenweg.


Wordpress is a feature full tool and nearly unparalleled as a blogging platform. By default, it has optimised search engine design including sophisticated permalink ability, spam filters, ability to tag posts in multiple categories, neat support for trackback and pingback standards and a very robust plugin architecture teamed up with a vast repository of plugins available making it a tweaker's heaven.

Working with WordPress on the move is also a no pain affair as native applications are available for most common platforms including but not limited to Android, iPhone/iPod Touch, and BlackBerry. Wordpress is used as an indispensable tool by a countable infinity of people, to name a few you wouldn't expect to be on the list: Central Intelligence Agency, Defense Intelligence Agency, Department of Education, Department of Energy, Department of Homeland Security, Department of State, Department of Treasury, Drug Enforcement Administration, Federal Bureau of Investigation, National Geospatial-Intelligence Agency and so on.

In this Fast Track, we will begin with the basics of WordPress and then move on to using Wordpress on its own hosted domain.

Once you have got your hands wet, we will move to installing WordPress on your server whether local or remote. Then we will move to managing your blog and keeping unwanted comments away. At this probably a text-only interface will get boring, so we will look into adding image galleries, videos, calendars and then blogging. We'll also learn how to manage multiple users. Lastly we will cover the advanced features such as feeds, syndication, podcasts, widgets, plug-ins and using WordPress as a complete content management system. It might be a little bumpy ride with tweaking of configuration files but hang in there and it will be fun and rewarding to no less extent.

IMPORTANT: We've included the last chapter of this Fast Track as a PDF in this month's Mindware DVD. We've done this as a test, so make sure to send us feedback about it to editor@thinkdigit.com.

It is important that you make your opinion known. We've got a lot of feedback telling us that a soft copy of the Fast Track in an interactive format on the DVD would be better than printing it on paper, so whether you agree or disagree, make sure to write in to the editor and voice your opinion. 

1 INSTALLING & CONFIGURING WORDPRESS



WordPress is one of the simplest open source blogging platforms to install. It has the famous “five-minute install” to its credit. In fact, installing WordPress can take a lot lesser than five minutes if you are familiar with the creation of a database on your web site. This process can be nearly instantaneous if you’re a power user who can script the database creation, or a minute or two if a web host control panel login is required. Most web servers today are already set up out of the box to handle WordPress. If you have PHP 4.3 or higher, and MySQL 4.1.2 or higher on your server, then you will not have any major issue installing WordPress.

1.1 System requirements

The only minimum requirement for installing WordPress is the PHP scripting language and a web server. The recommended system essentials are PHP 5, MySQL 5, and Apache 2 on a Unix-variant operating system such as the ones most hosting companies provide in their Linux packages.

The foundational building blocks of WordPress are the PHP scripting language and the MySQL database. Though it is technically possible to run WordPress on a different database than MySQL, it is impossible to run WordPress without PHP.

Perhaps the most important area for system optimization is the MySQL database, at large scale. Some of the most intensive activity in the WordPress system architecture occurs here. Out of the box, MySQL is not optimised for heavy WordPress. It works well only with small blogs. After that, it begins to bog down.

1.2 Choosing a web server

In most cases, the choice of web server is something that is pretty standard. Most web hosts provide Linux hosting with Apache as the configured web server. In most cases, the choice is automatic because it is the web server that comes with virtually every distribution of Linux and is turned on, almost universally, by default.

With rare exception, these hosts are perfectly fine out of the box. Little additional configuration is needed to run a small blog. However, you should be a little cautious and ensure that the host has the minimum software

requirements (PHP 4.3 and MySQL 4.1.2) and that they have the `mod_rewrite` Apache module configured.

1.2.1 Apache

Apache is an open source web server that comes bundled with most Linux distributions. It is, by far, the most commonly used web server, and WordPress was designed perfectly for it. There is no minimum requirement for WordPress and Apache. The oldest version of Apache currently in use is Apache 1.3 and it works as well as Apache 2.2, which is the most stable version.

1.2.2 Internet Information Services

Because PHP can be loaded into the Microsoft server product, Internet Information Services (IIS), WordPress can be run on it. IIS is not the optimal solution, but it works in scenarios where other server products are not available or permitted. IIS is bundled with Microsoft Server products as part of Active Directory. The trickiest issues with WordPress on IIS are permalinks.

On Ubuntu, the demonstration platform here, the previous command makes `mod_rewrite` available to Apache but it won't enable it. To do that, you need to make sure Apache knows how to load it. Follow these steps to create a symbolic link (better understood as a shortcut to Windows users) and restart the web server:

1. From the Linux command line, type `ln -s /etc/apache2/mods-available/rewrite.load /etc/apache2/mods-enabled/rewrite.load`. If you're using Apache 1.3, simply replace `apache2` with `apache`.
2. Type `/etc/init.d/apache2 restart` for the changes to take effect.
3. Type `apache2ctl -M` to verify that `rewrite_module` is in the list of available Apache modules.

The nice thing is that it is highly unlikely you will need to do this. On other variants of Linux, it is much easier to enable `mod_rewrite`. Ubuntu and Debian modularise the Apache configuration file (sometimes called `apache2.conf`, but typically called `httpd.conf`) and break the portions of the configuration into separate files. Usually, it's one big file that can be edited. The key bit that Apache needs is a line that says `LoadModule rewrite_module /usr/lib/apache2/modules/mod_rewrite.so` where `/usr/lib/apache2/modules/mod_rewrite.so` is actually the path to the compiled `mod_rewrite.so` file.

1.3 Installing WordPress

Now that you have all the elements in place to run WordPress efficiently, PHP 4.3 or better, MySQL 4.1.2 or better, Apache or another configured Web server, and perhaps some optional caching or communications modules. The file size for a copy of WordPress is intentionally small. The total size of the ZIP file for WordPress 2.8 is 2.2MB. You can download the file at <http://wordpress.org/latest.zip>. Then unzip the file to find the `wordpress/` directory.

1.4 Configuring the database

When you installed MySQL earlier, you didn't create any databases. Most servers have some sort of Graphical User Interface (GUI) for doing this. Most of these GUIs use a web-based software called phpMyAdmin for managing databases.

If you don't want to use phpMyAdmin, you can create a database from the command line as shown below. The first command creates a new database called `wordpress`. The second command creates a new user called `wpuser` and sets the password to `wppassword`. It assigns the new `wpuser` to the `wordpress` database (and all tables in the `wordpress` database) and it gives it full access to that database, as long as the user is trying to access the server only from the same server (`localhost`). In some rare cases, your hostname will be different than `localhost`, but generally, for a single WordPress install on a single server, `localhost` will do.

Creating a MySQL database, username, and password for WordPress

```
[root@wpftdigit ~]# mysql -uroot -p
Enter password: <enter root password>
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67-0ubuntu6 (Ubuntu)
Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.

mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* TO wpuser@
localhost IDENTIFIED BY 'wppassword';
Query OK, 0 rows affected (0.00 sec)
<enter Ctrl-D to exit MySQL>
mysql> Bye
```

Now that you have created a new database to use, you have to tell WordPress about it. In the newly unzipped `wordpress/` folder, you'll find a file called `wp-config-sample.php`. Copy this file and rename it to `wp-config.php`.

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wordpress"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="username"/>	Your MySQL username
Password	<input type="text" value="password"/>	...and MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if <code>localhost</code> does not work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Your gateway to WordPress

1. Replace `define('DB_NAME', 'putyourdbnamehere');` with `define('DB_NAME', 'wordpress');`.
2. Replace `define('DB_USER', 'usernamehere');` with `define('DB_USER', 'wpuser');`.
3. Replace `define('DB_PASSWORD', 'yourpasswordhere');` with `define('DB_PASSWORD', 'wppassword');`.
4. If your database host name is anything other than `localhost`, replace `define('DB_HOST', 'localhost');` with `define('DB_HOST', 'newdbhostname');` where `newdbhostname` is the name or IP address of the database server. This is optional and likely not required.

There are some additional optional settings in here as well. Notably, it is a good idea to change the `table_prefix` setting. It can be anything you want and defaults to `wp_`. Additionally, four other settings are defined, `AUTH_KEY`, `SECURE_AUTH_KEY`, `LOGGED_IN_KEY`, and `NONCE_KEY` that should all be changed to unique strings or phrases. They can be anything you want, and you'll never have to see them again. They simply help WordPress operate securely. You can auto-generate random settings for these four secure keys by visiting <https://api.wordpress.org/secret-key/1.1/> and copying and pasting the result on top of the exiting default settings. An example of randomly generated secure keys from <http://api.wordpress.org/secret-key/1.1/>

```

define( 'AUTH_KEY',      '/~*:GN6QJE1,N^HayyNc@47?ug[])M|s) #
5(i{%W/
VmvDlVhn[GU@yaN[X(uH|@x') ;
define( 'SECURE_AUTH_KEY',      '5rXNA}tA#7DCm{=?EN-L/
b2=3&7@4wPOfsJ<$]
fIim=V=yIb(p]<t3k,80A-!Uz!') ;
define( 'LOGGED_IN_KEY',      '3$+S?|mun`&GT2]||/,) .
I7X4s,&HR^V,%*nZ<j<Bg4|P-
*I;+_!jn7>f@||;8`Pg') ;
define( 'NONCE_KEY',      'C-&mE>EmyLs^D_ZA,(Y+63W+G1UL)
ZjS+,u-/@
ubnu,z{j,/P#44gLW?Q+|$(Y. ');

```

1.5 Uploading your files

Once you have finished editing `wp-config.php`, it's time to upload your files to the server. FTP (or better yet, SFTP) the contents of the `wordpress/` folder to your new web space. It will take a minute or two, but once that is complete, load your site in a browser. If everything is configured properly, you should see the first page of the installation wizard.

If you see a database error, go back and double-check that the database credentials you supplied in `wp-config.php` are correct. Once you see this page, follow these steps to install WordPress:

1. Type the name of your blog in the form field labelled Blog Title. For this demonstration, we've used "The Fast Track to WordPress."
2. Enter your e-mail address. This will become the general e-mail address that is used for administrative purposes on the site. It can be changed later.
3. Click the Install WordPress button.

There is only one step. The next page provides a randomly generated password for a new user called `admin`. This information will also be sent to the e-mail address you designated during installation.

After installation, WordPress generates a randomly generated password for the new user called `admin`. With the new username and password in hand, it's time to log in to WordPress for the first time. **d**

2 MANAGING A BLOG

Starting a blog might seem like one of the simplest things to do. After all, all you need is a web page that you can type your thoughts out on. Earlier, when blogging wasn't so prolific, people had to create their own HTML pages, find a web site that would be kind enough to host it for them (at a nominal cost, or preferably free) – (remember the days of Geocities) and then try and make the blog interactive. Because a blog's just not about the content anymore, it's also about the packaging. This is where WordPress scores. For those of you who found the earlier chapter slightly mind-boggling, the best alternative is to zoom over to <http://www.wordpress.com>, (it's wordpress.org for the downloadable version which you'd like to host on your system). Create an account and start blogging on a blog hosted by WordPress. In terms of usability, there is no difference between wordpress.com and wordpress.org. The interface is identical.

2.1 Signing up for a blog

Signing up is ridiculously simple. You're asked to create a user name, protect it with a password, and provide an e-mail address, for mails, verifications and other formalities. Of course, the mail is also how you are notified of posts and comments made on your blog. Sign up for your blog while you're at it.

If you're too unsure about the name, don't be, there's nothing limiting you to only one blog, and many bloggers have been known to change the names and domains of their blog on a whim. One of the good things about WordPress is that it's completely open, and there's no lock-in. That is, assuming you are dissatisfied with Wordpress and would prefer Tumblr, Posterous, Blogger, or any other service, you can export everything to an xml file and import it later on to your new blog.

Choose a blog name and you are good to go. While signing up, make sure you check the box for "I would like my blog to appear in search engines likes Google and Technorati, and in public listings around WordPress.com". Though you have a long way to go in case you want to improve search optimisation, this is a must in case you want your content to be discoverable. Technorati searches blogs and also rates them.

2.2 Creating a profile

A profile is important. Depending on the content of your blog, your

authenticity as an author matters. Though anonymity is accepted and often appreciated on the internet, it does help if you mention your credentials, or your areas of speciality and interest. Also, in the case that you do plan to use your blog for promotional purposes, especially for promoting yourself (as an employee or similar) you can upload your resume to attract potential employers.


You should receive an email to authenticate your WordPress account activation. Also provided should be a link to your blog on your email. Depending on the name you provide, the location of your blog will be `http://<enter name provided>.wordpress.com`.

2.3 Your first blog

WordPress automatically creates a blog for you with a default entry, called “Hello World” (Take any class on computer programming, and chances are this is the first thing they teach you to output). It also chooses a default template. Through the course of the fast track, you’ll figure out how to

Thriddas's Blog

Just another WordPress.com site



[Home](#) [About](#)

Hello world!

Posted on [July 29, 2010](#) by [thriddas](#)

Welcome to [WordPress.com](#). This is your first post. Edit or delete it and start blogging!

Posted in [Uncategorized](#) | [1 Comment](#)

Recent Posts

- [Hello world!](#)

Archives

- [July 2010](#)

Categories

- [Uncategorized](#)

Meta

- [Register](#)
- [Login](#)
- [Entries RSS](#)
- [Comments RSS](#)
- [WordPress.com](#)

The default template created by WordPress

customize the aesthetics of your blog to your taste and how you can even create your own themes.

Now, to create your first blog post, go back to www.wordpress.com. On the top, there are fields to enter your name and password, which will take you to your account.

Click on My Dashboard, to be taken to your base of operations. This is the place from which you manage your blog posts, edit comments, manage your themes and widgets.

If you were tempted to check out the My Account, this is a explanation of the list of things that appear when you roll your mouse over.

1. Global Dashboard: Has other WordPress blogs, a few stats, and of course, your blog details. It should have a What's Hot (trending blogs and topics), Top WordPress.com blogs for the day, Top Posts, and Fastest Growing Blogs among others.

2. Stats: WordPress.com automatically provides you stats once you create your blog. Though you can sign up for more advanced statistics using Google Analytics or similar software, your basic WordPress statistics are good enough for the average user. It shows your top posts, the number of views your blog has got over the last few days, weeks, and months. It also has the list of sites that send you traffic. Of course, these are things you should be considering once you've started blogging actively.

3. Likes: You can choose to like a WordPress.com Post by clicking Like (at the top of most blog posts).

4. QuickPress: This is for those who want to pen down their thoughts quickly, and really don't want to waste time formatting. Just choose a title, type down the content, choose the tags, and publish

5. Blog Surfer and Tag Surfer: You can view other blogs of your liking, alternatively you can choose and add blogs based on 'tags'.

6. My Comments: Allows you to manage comments.

7. My Blogs: For all of you ambitious people who choose to have more than one blog. You can manage them all from here.

8. Edit Profile: Edit the information you want to share with the world.

Now, you need to focus on the dashboard of your blog.

Quick Tip: In case you don't feel like going to WordPress.com, Signing In, Clicking on My Dashboard, a quicker way is to go to `http://<enternameofblog>.wordpress.com/wp-admin`.

Below is a screenshot of what your blog will look like the first time.

Please note that all those blocks of content are frames which can be moved around and customized to suit your aesthetic.

The “Right Now” header for example shows you the state of your blog at a glance, listing the number of posts, pages, comments, categories and tags.

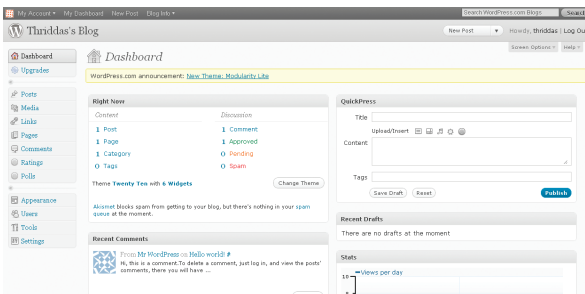
Your current theme is also mentioned. Click on “Change Theme” to select from 99 themes that WordPress has.

2.4 Creating a blog post

Let's now get down to the essence of content creation. You can, of course, use the QuickPress option (also available on your main page in the form of a header) to quickly create a no-frills, no-nonsense blog, but that belies your entire intention of using WordPress.

Title

The Title is something you must pay attention to. Apart from your grammar or journalism teachers telling you how the title is what defines the story, draws attention to the content, in WordPress, by default – the title of your post is the URL.



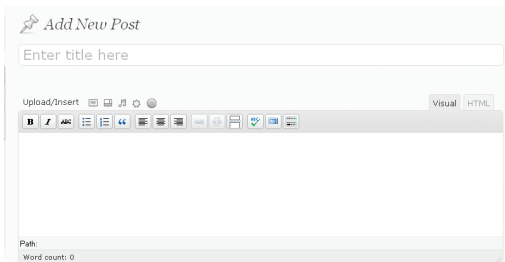
The above is the default template created by WordPress

The default Permalink (Permanent Link) to your blog post would be: `http://<blogname>.wordpress.com/<datein-yyyy/mm/dd-format>/<title>`.

The title, of course, for those of you fiddled with the permalink is separated by hyphens.

Blog Content

Well, blogging is not just about typing down a few random thoughts that come to your head. Today, with the advent of social media, UGC or User Generated Content is very crucial to the stuff you see on the Internet. Of



Click on Posts, and click Add New on the drop down to start working on your first blog post

course, since this FastTrack is not about writing skills or blogging etiquette, we shall focus on HOW to blog, and not on WHAT to blog.

Since we're not focusing on content, focus instead on formatting. WordPress comes with a minimalist formatting toolbar, which might seem limited to those of you who use Office (Word) or any other Document Editor.

The formatting toolbar is all that you'll need to stylize your blog post.

This might seem slightly redundant, but let's go over each button:

1. B – Bold – Emboldens the text
2. I – Italics – Italicizes the text
3. ABC (Strikethrough) – We have often wondered how prevalent is this need to type out stuff, and strike it out, as opposed to merely deleting it.
4. Bullets – Very useful while listing points
5. Numbering – This however, comes with no auto-formatting. Do not type 1. and expect WordPress to infer
6. Quotes – Useful for showing statements quoted verbatim, also for emphasis
7. The next three are alignment – no explanation required
8. The next symbol is for the hyperlinks, do not confuse it with the 'attachment' symbol. Hyperlinks, now commonly called links are very essential for blogs. For our purpose it's sufficient to know that it helps link to other pages. If you've quoted something and you'd like to provide the source,



The formatting toolbar is all that you'll need to add formatting to your post

or if you just want someone to visit some funny site, a hyperlink is the way to go. Select the text you'd like to make “clickable”, then click on the hyperlink button. You'll see a box allowing you to enter the URL (type in the complete URL with http://).

Type in the Title (this also serves as alternate text when you roll the mouse over the link). A good practice is to open the target in a new window. This ensures that the reader does not get carried away to the other blog/web page and forgets about yours.

9. The symbol next to the hyperlink is the un-link symbol, used for removing links. The Insert “More” tag is good for long posts. In case you don't want your blog to be full of long-winding articles, you can add this. What this does is that it will only display the part above the “More tag” on the blog. Clicking on the Show More will lead the reader to a new page where the post is shown in full.

10. ABC checkmark is the Spell Checker. Though most people think that the Internet is a place where you can get away with poor spelling and bad grammar skills, it does reflect poorly on the owner of the blog in case your posts are cluttered with sentences confusing ‘their’ and ‘there’. As advice, even if you do let other people write on your blog (we'll discuss accounts later, please go through their work and edit for errors)

12. The next button toggles the full-screen mode.

- a. Advantages: More space for writing, viewing, editing
- b. Disadvantages: The rest of the dashboard goes missing

Now, you might think this to be fairly limited. On our first use of WordPress, we were amazed at the lack of a “Justify” alignment button. For those who wish to add more formatting, there's the kitchen sink button (next to the full screen mode).

The Kitchen Sink has more commonly used editing tools like the quintessential undo and redo ([Ctrl] + [Z] still works).

It has also has options for:

- 1. Different styles (preformatted)
- 2. Underline (again, Ctrl+U)
- 3. Justified text
- 4. Font Colour

5. Paste Options

6. Remove Formatting option (use this when you paste from different sources. When copying text from Word or webpages from the internet, you'd notice that WordPress does not often copy them correctly. Use the Remove Formatting option and format it yourself)
7. Insert Symbols (and custom characters), though a person with decent knowledge of Unicode could manage well without it
8. Indenting for formatting your content
9. The famous un-do and re-do options
10. Help function

a. Shows basic Rich-editing help

b. Also has the Hotkeys (Shortcuts) which can speedup your formatting skills

This is however, limited functionality for a blog. Especially now, with multimedia being more popular than textual content. To upload images and videos to your blog, see the buttons above the formatting toolbar.

You have an option to either upload from your computer, or to add an image from a URL. The advantage of uploading a file from your computer is that it gets stored in the media library, from where you can re-use easily. Apart from accessibility issues, the image also loads faster if it is uploaded. Of course, there is an upload limit.

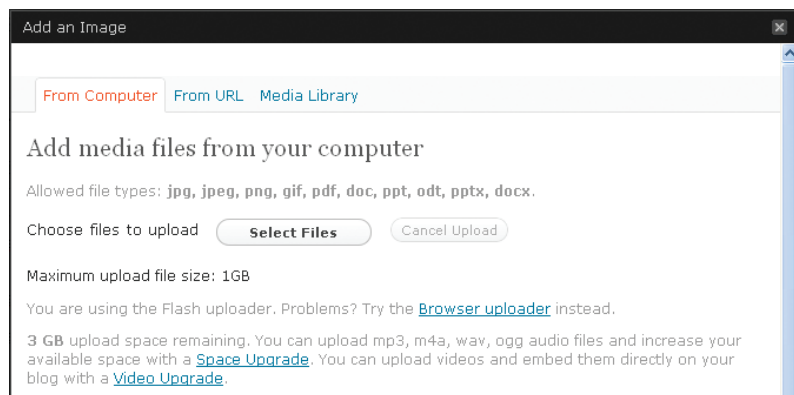
Once uploaded, you can resize it, make it a link, have alternate text, or change the placing. The same is true for other uploaded multimedia. You can modify sizing, positioning to suit your convenience. Wordpress allows you to upload audio and video. Of course, for additional functionality like the ability to embed YouTube / Vimeo videos or Scribd documents, you can add a few widgets.

There is additional functionality for adding “other media” which includes Office Documents and other files. You can also create polls, (for which you'll need to create an account with PollDaddy.com – a sister concern of WordPress.

For those who think this is limited functionality, there is always the additional option of html based formatting. If you look at your blog, there are two options, Visual and HTML. The Visual presents your formatting in the standard WYSIWYG (What you see is what you get), while WordPress converts it to HTML format. For those who understand the language, you can check out the HTML option which allows you to specifically tweak and format your content.

Now that you've figured out a way to add some content to your blog,

the obvious choice is to publish it. But before you do that, you need to add a few things.



Add images and even video to your blog

The key points you need to consider are:

1. Tags – Tags are essential to any blog post. For one, it gives a very clear indication of what the blog post is about using keywords. Assuming this was on our blog, a few tags that could be used would be “FastTrack”, “blogging”, “wordpress” and so on. Over time, experienced use more or less the same tags. The advantage is that tags are linked. So you can see all blog posts under that tag. For your WordPress blog, that link would be : <http://< name of blog >. wordpress . com / tag / <tag name>>. Remember to separate tags with a comma while inserting. We recommend short simple tags that you will use commonly as opposed to long winding sentences. Please remember a tag is exactly what the name suggests, not a summary of the blog.

2. Categories – This is for all those people who want their blog to be organized. This is rarely used for personal blogs (unless you’re one of those who insist on order and staying organised), but could still be useful. For example, say your personal blog is divided into

- a. Your personal experiences
- b. Rants on the government
- c. Your Views on <insert topic of choice>

You could create three categories that would help people read specific content. This, of course, works better if you are hosting a tech blog or similar, because then you have a wider range of categories that your blog posts belong to like “Mobile, Internet, News, Views, Reviews, Gaming” etc. Again, posts in the same category get clubbed together. So, your standard link would be `http://<name of blog>.wordpress.com/category/<category name>`.

You can also create sub-categories (as you can see in the screenshot attached). Select New Category, type in a name, and in case this is a sub-category, choose the appropriate Parent Category.


3. Publishing – Wordpress gives you quite a bit of freedom when it comes to publishing your posts. You can choose to publish it instantly, alternatively you can schedule it for a later date. You may also save it as a draft. When we come to multiple users at various levels of access, you’ll understand the importance of publishing options. You also have the option of making the blog private, public or password protected. In case you wish to make it public, you also have the choice of publishing it on the first page.

Before publishing, an act of sensibility must prevail. There is an option to preview the post. This is recommended because on several occasions, addition of media, and pictures might ruin your text layout. Certain aesthetics will only be revealed once you publish. Hence a safer choice is to preview it.

Check your main Posts page now. You should be able to see a list of your recent blog posts. The posts page is extremely useful. For one, you get a quick glimpse of all the posts, the number of comments that need your attention, the author, the categories and other basic information. A quick-edit also allows you to correct errors that might have crept in, like misspellings, wrong tags etc (Roll your mouse over the blog post to find the quick-edit option)

So much for writing and editing.

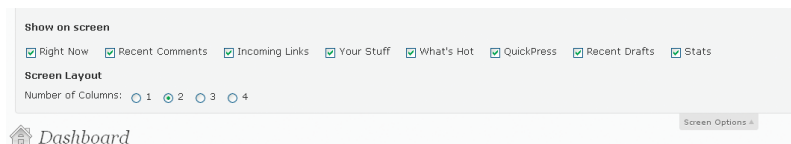
There’s just one more topic which is essential to a WordPress blog that features in the writing category.


WordPress allows you to create pages to add to your blog. Those familiar with blogging will realise that often, there is a need for an “Introduction to My Blog” or an “About Me” or “A list of Sites that I enjoy” or similar. Click on Pages. There’s already one created by default for an About Me. You can add more pages. Unlike blog posts where the most recent one is all that the viewer sees, the pages are always accessible. 

3 OPTIONS, COMMENTS & ACCOUNTS

Now that you know the basics of adding posts to your blog, you can move on to customising to suit your preferences. Here, we deal with the most common options you'd want to tweak on your blog, comments and related settings, and how you can work with different accounts and authorize different levels of access.

Let's begin with a few easy customization options. Don't like the widgets you see on your dashboard as part of your opening screen? Notice the Screen Options on the top right (below the friendly Howdy). It gives you the option of frames you want displayed, and the number of columns you'd like. You can move the frames around by clicking on the header and dragging.



 *Dashboard*
Absolute control from your Dashboard

Let's focus on the Blog Options first. Click on Settings.

Settings has the most basic options. It allows you to set the following:

- The Title of Your Blog (this is different from the web-link for your blog)
- The Tagline – a one line description for your blog, make it catchy.
- The email provided is used for administrative purposes. So, all changes to your blog that you need to be notified of will be sent to this one. In case you want to change it, WordPress will want to verify your identity at your new address before confirming the change
- Your timezone is important. For one, it puts things in perspective for international readers (if any). Also, its essential for basic time settings
- Date formats are a matter of choice. Some prefer the short crisp dd/mm/yyyy, others prefer to cater to American tastes and go mm/dd/yyyy. You can also set your own date format
- The same applies to time
- The day the week starts on (mostly it's Sunday or Monday – depending on local preferences – We personally like to think Monday, because then you have a nice rewarding weekend at the end of it).
- And the Language settings

As you must have realised, these settings are pretty basic. However, if you notice carefully, these are but the General settings. Your settings option, just like your Dashboard or Posts option has a dropdown that allows settings for

- a. General
- b. Writing
- c. Reading
- d. Discussion
- e. Media
- f. Privacy
- g. OpenID
- h. Domains
- i. Webhooks

We shall quickly go through each one of them

Writing

The following options are available:

a. Size of Text box: This is reflected in the text box that's available to you for typing content when you create a new page / blog post. Default setting is 10 lines.

b. Options for emoticons (recommended) and correction for XHTML errors (up to you). You really don't want Wordpress correcting presumed XHTML errors when there are none (or you intend it to be that way)

c. The default category is Uncategorized. Since you can set this setting based on post-specific content, it really doesn't matter what you set this as

d. The bookmark called "press this" is to be added to your bookmarks toolbar on your browser. A lot of people might not like their browser cluttered with toolbars thereby making this feature useless. This allows you to add content directly from an external source. You can clip text / media (pictures and videos) directly to your blog. Add this if you spend a lot of time writing on interesting stuff you find online

e. The post by email feature is useful for blogging on the go. This allows you to send a blog post as an email to WordPress – pretty useful for those who wish to blog while traveling. After all, an email on your mobile phone is easier to type in (plus it's offline) as opposed to using the WordPress mobile interface (<http://m.wordpress.com>). To enable this you'll need to go to your Dashboard > My Blogs and enable Post by Email. This creates a very wacky email id for you. Keep hitting regenerate till you find one you find

acceptable. It is an @wordpress.com email address. Please note that there are certain tags you need to insert and certain formatting details you need to take care of. A simple email does not cut it. Check <http://en.support.wordpress.com/post-by-email/> for proper tips and formats

f. While we're at it, if you do get tempted to enable the Post by Voice feature, gladly do so. But do understand that it's an International number that you will be calling. You may record up to an hour. But considering international dialling rates, you'd probably want to skip this feature

Reading

a. You can now set your start page to a Page that you've previously created. The default setting is to display the latest blog post(s).

b. You can choose the (maximum) number of blog posts that appear and the number of items on RSS feeds. We will be dealing with syndication and feeds later.

c. You may choose Full Text v/s Summary Options to be viewed in a feed. A lot of blogs don't mind displaying the full article. It's very convenient for the reader. However, if you have a lot of interactive content on your blog (also sometimes called ads) that you'd like to engage the user with, you might want to just give them a teasing glimpse of your content.

d. You also have the choice of adding extra information in the RSS feed like adding to social bookmarking sites, and categories and tags.

e. Ignore the encoding, unless you're really feeling adventurous and feel like experimenting

f. An invitation text is sent out in case you decide to invite people through your blog. Please change the text to something a little more personalized / simple instead of the "Howdy"

Before we come to the discussions sub-section, you need to understand what a comment ideally is. A blog is mode of communication, and any communication is rendered effective only with a reply, acknowledgement or criticism. Comments are your way of knowing that people read your blog and actively engage with you. Check any leading author in the blogosphere today and every post will be numbered with comments. Often, a blog becomes less of a medium of interaction between author and reader and more a forum for people with like interests. It is these fora that allow niche groups to discuss their thoughts on topics of similar interest.

You obviously have a choice to disable comments on your blog. When writing a new post, notice the Discussions block below the text box. Uncheck

the “Allow Comments” box to disable comments – this isn’t recommended, though it’s a personal choice.

Discussion settings

a. Your default article / post settings are visible which basically allow comments, pingbacks and trackbacks.

‘Comments’ is easily understandable. You’ll need to read up on pingbacks and trackbacks if you wish to fully understand the two (there is documentation available on the web). A nice simple explanation of a “trackback” available on the web was this :

- Person A writes something on their blog.
- Person B wants to comment on Person A's blog, but wants her own readers to see what she had to say, and be able to comment on her own blog
- Person B posts on her own blog and sends a trackback to Person A's blog
- Person A's blog receives the trackback, and displays it as a comment to the original post. This comment contains a link to Person B's post

Pingbacks are similar, except they don’t allow content (like the author’s own comments) and instead only have the link. They use a different protocol and have a layer of authenticity. Most of us just ignore this. Ensure you have pingbacks and trackbacks enabled, the rest WordPress will take care of.

b. As we discussed earlier, the above can be overridden for individual articles

c. Usually, most people don’t prefer Anonymous comments. Though anonymity on the web is useful and often essential, you may wish to know the identity of the commenter (or a way to contact him). Hence, the author must fill out a name and e-mail address (of course, if he really wanted to, he could fake both)

d. Wanting users to register does not make sense for a personal blog. For that matter, it makes commenting an extremely tiresome process which might lead to your readers not doing so.

e. You can disable comments on older posts after a fixed period of time

f. Nested comments help with conversations. If someone has commented on your post, and someone replies to that comment, it creates a mini-conversation of sorts. Now, someone can comment on the comment. You may choose how deep this runs. Sites like Reddit have layers of conversations which often leaves you wondering what started the entire comment. This is an example of how blogs act like ad-hoc forums

g. Depending on how much you care about your blog and the type of

comments that come in, you may choose the notification settings as well as the comment approval settings. I would recommend a notification every time a comment is made, irrespective of whether it is approved or not. (Check both). With regards to approval, the usual method is to allow the moderator to check whether it is valid and then authenticate it. Though Akismet does a decent level of filtering, a lot of people still use manual blog marketing to insert an inane comment and then advertise their sites.

h. Alternatively, you may choose to approve all comments by an individual whose comment has been accepted before.

i. You can also set up filters based on comments containing certain words or links to certain sites. These will come up for moderation. You can limit the number of links or choose inline text filtering. Please note that WordPress searches within the text and does not do whole word matching. So if you're filtering for "sex", a comment having "sensex" will get flagged for moderation

j. You can also set up filters for spam

k. You may choose to allow comments via email, allow a person to subscribe to the comments (in case of a really interesting conversation with several comments) and an option to subscribe to the blog.

l. Now, an Avatar is an identity given to the user based on his email ID. I will not go into the whole MD5 encryption and other details, but it's sufficient to know that it creates a nice icon next to the commenter. Depending on the WordPress plugin that you use for allowing Avatars, a picture will appear next to the commenter. A Wavatar generates a funny cartoon, a Gravatar is a Globally Recognized Avatar (you can create one for yourself with your picture etc). Check the tag next to the option to check whether the user needs one, or if your blog will generate it.

m. You have the choice of allowing the Maximum Rating. (This is limited to the Gravatar option – if you don't want Gravatars with explicit pictures)

Media Settings:

Media Settings require little or no explanation.

a. To those of you who tried inserting a picture in the earlier chapter, you would know that you have a choice of inserting the picture as-is, as a thumbnail, medium size or large size. This allows you to configure those sizes in pixels. (You can of course, change this once the picture has been inserted. Hover over the picture, top left has options. Click to resize by percentage)

b. Allowing Auto-embeds is useful. This allows you to embed videos, pictures, documents from various websites. Just leave the default settings on

and try it out. Go to your blog and paste the hyperlink of the page you want embedded. For example, to embed a Youtube video, go to the video, copy its URL and paste it. A few rules. One, it must be on a separate line. Two, it must not be a hyperlink. Auto-Embeds work only for a select few sites pre-decided by WordPress. Popular ones like Youtube, Scribd, Flickr, Vimeo are present. The full list is here: <http://codex.wordpress.org/Embeds>.

c. Choose the Free Video Formats if you're a stickler for Open Source and have principles and views on free video formats and those that charge royalty.

Privacy settings

These allow you to set the visibility of the site. Best leave the default options as they are.

OpenID allows you to log in to other sites that support the OpenID standard. If you're active on the web, you would know that there are a lot of sites that require you to register, and sign on in order to comment or participate in discussions (among other things). If they support the OpenID standard, you need not create a new username and password. You can just use your OpenID. This saves you the hassle of remembering hundreds of usernames and passwords that you might have to create. In this case, your OpenID is the URL of your WordPress blog. (It's like Facebook Connect – if you want a good parallel)

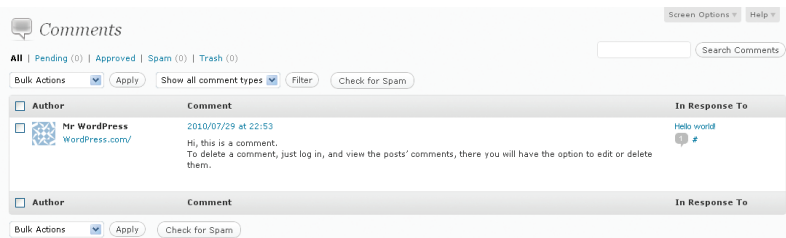
In case you own a separate domain and would like people to be directed to your blog, add the domain name in Domain (under Settings).

As for Webhooks, you can use them to trigger events. It's slightly complicated functionality, so we'll ignore it from this FastTrack.

That's it for the settings that you need to worry about.

Once the comments start pouring in, you might want to check the Comments to see which to approve and which not to. Also, while running a blog, it's essential to monitor the comments from time to time, you really don't want a lot of old comments unapproved. Here, you have the option of deleting, approving or marking as spam. In case the comment is worded badly, you also have the choice of editing the comment. If you can see small speech bubble below "Hello World", that lets you know the number of comments. In case you have some comments waiting for approval, it is coloured.

Coming to Accounts and Authors, your blog need not be managed by you alone. Several blogs are the results of collaboration. Assume you have just



Moderate the comments on your page from the Comments panel

started a blog on technology, and you'd like other tech aficionados to author a few articles. Since you can't give them your login ID and password, and you really can't wait for them to mail you the articles, format it and then put it up (too much trouble). Alternatively, you could consider creating a new user who can use the blog.

Click on Users. You should be able to see a list of users who are authorized to post on your blog. As of now, you are the sole user registered with admin rights!

To add a new user, just type in the email address of the new user, select their role (this defines their level of access), and you're good to go. The new user needs to have a Wordpress.com account, else you'll be prompted to invite them.

Therein lies the difference between Wordpress.com and Wordpress.org. Wordpress.org allows you to create a new user with some level of access, and you as the administrator set the initial password (which they modify later). In Wordpress.com, they need to have an account, and you have no access to their password.

The roles in Wordpress.com are as follows :

There are four roles for users that you wish to add to your blog: Administrator, Editor, Author, and Contributor.

Administrator: An administrator has full and complete ownership of a blog, and can do absolutely everything. They (basically YOU) have complete power over posts/pages, comments, settings, theme choice, import, users – the entire packaged deal.

WordPress recommends having only one administrator per blog. They also issue a disclaimer that if you do give someone administrator rights and he decides to delete the entire blog, he's quite entitled to do so.

Editor: An editor can publish, edit, and delete any posts/pages, moderate comments, manage categories, manage tags, manage links and upload files/images. Editors also have access to the blog stats. Basically, he has complete control over anything that is relevant to content, but no control over the administration aspects like users or appearance.

Author: An author can edit, publish and delete their posts, as well as upload files/images. He can manage comments which are relevant to HIS post alone. Note the difference: While an editor can moderate comments across the blog, the author is restricted to posts created by him.

Contributor: A contributor can edit their posts but cannot publish them. When a contributor creates a post, it will need to be submitted to an administrator for review. Once a contributor's post is approved by an administrator and published, however, it may no longer be edited by the contributor.

Please note that a contributor's post can also be edited / modified by a person with Editor Capabilities. A contributor does not have the ability to upload files/images.

Wordpress.org has the additional feature of having a Subscriber feature.

A subscriber basically subscribes to the blog, and can have a profile on the blog. He could be a regular reader, an active commenter, or just a huge fan (or all of the above).

In addition, you can modify the above rights in the Settings > General Panel. The default option can be changed here. For deleting / modifying account access, check the Users Panel. Please note that only the administrator can make these changes or even view the Settings Option. It is denied even to user accounts with Editor status.

For those of you who plan to manage a Wordpress.org self-hosted blog, you should know that there are widgets available that can customize the level of access. For example, if you want your editor to be able to modify the theme, or imagine some strange democracy where even subscribers can moderate comments, there's a widget that helps you do that.

Now that we're done with Access Levels, you might want to add a more human touch to your blog. As a contributor, you might want others to know who you are. Your personal profile is in Users > Profile. You could of course, reach the same place by clicking on the friendly "Howdy, <user>" that appears on almost every WordPress page. Your user name is a hyperlink.

Once you fill in personal details (your name etc. would already be filled in) like phone number, IM nickname, location, you've fully threshed out your profile.

Now, check Personal Settings (under users). In the case that you're an avid blogger and often blog on-the-go, you might want to enable Geotagging. This will publish the location you are blogging from.

The Personal Settings has a lot of options, like a Fun Mode. It's a recent addition (WordPress announced it on its blog in April this year) and WordPress encourages you to try it. Since they don't reveal it, we'd like to play along. Figure out for yourself what Fun Mode really does.

You have options for colour, enabling Keyboard Shortcuts, accessing the Twitter API and allowing blog posts via Twitter (assuming you actually think microblogging is the same as blogging).

This is something we'd personally encourage Proofreading. Enable proofreading on your blog. Please note that in the case of a personal blog, this might get annoying with it pointing out all the proper nouns and strange street names.

Automatically proofread content when:

- ☐ a post or page is first published
- ☐ a post or page is updated

English Options

Enable proofreading for the following grammar and style rules when writing posts and pages:

- ☐ Bias Language
- ☐ Clichés
- ☐ Complex Phrases
- ☐ Diacritical Marks
- ☐ Double Negatives
- ☐ Hidden Verbs
- ☐ Jargon
- ☐ Passive Voice
- ☐ Phrases to Avoid
- ☐ Redundant Phrases

[Learn more](#) about these options.

Language

The proofreader supports English, French, German, Portuguese, and Spanish. Your [blog language](#) setting is the default proofreading language.

- ☒ Use automatically detected language to proofread posts and pages

Ignored Phrases

WordPress can take care of your language

You can also change your account details like e-mail and password.

That's it for the settings. Now that you've figured out the basic configurations of WordPress and how to customize it for your requirements, I'd advocate a lot of healthy tinkering. Modify things, try editing things, use widgets for more access and better control, and if you really like a challenge, get a Wordpress.org account and try the same on a self-hosted blog to figure out just how much more control you have. **d**

4 THEMES

In WordPress, themes are a major part of the user experience. As with any web site, the user interface can make or break a site's capability to build traffic, expose content for discovery, and enhance user stickiness.

Finding the balance between these three areas is a somewhat fuzzy, experimental process. Most WordPress users often change their themes. Some do it to experiment to see what works best while others do it to showcase their own design abilities and to try new things. Still others change themes regularly for no other reason than that they are bored.

4.1 Things to consider when selecting a theme

Typography

User experience experts strongly agree good typography is essential, but the opinions diverge from there. Some will say that using a serif font (like the popular Georgia font) sends a strong message of stoicism and tradition while arguments on behalf of sans-serif fonts (Arial, Helvetica, and so on) send messages of elegance and clarity.

Font discussion continues beyond font type to sizing, styling, words per line, line spacing, and so on. Every user-experience professional holds to basic principles of font selection, but many diverge on specifics because it tends to be a topic of personal taste.

White space

Another common consideration when selecting a theme is white space. White space tends to make a site more readable; but, similar to fonts, do not guarantee the site will draw users. One technique is to add vertical spacing between lines of text. Additionally, you can enhance white space with ample padding and margins around images, pull quotes, or sidebar spacing. When white space is not used with care, a site can become “cluttered” and visually chaotic. Readers tend to have little patience for this kind of experience.

Fixed or fluid width

A third element to consider when choosing a theme is whether the theme has a fixed-width or fluid-width layout. Fixed-width layouts have a specific width that never changes, regardless of the size of the browser window. Fixed-

width layouts often are designed to accommodate a specific screen resolution and can have too much white space when viewed on monitors that are larger, or with resolution that is higher, than specified. However, they are usually easier to design because dimensions and expectations are determined before the design starts and there is much less room for unknowns. A good example of a fixed-width web site design is the BBC news site at <http://bbc.co.uk>. As the browser window is resized, the elements of the layout don't change, even when the browser window width is smaller than the overall width of the page.

On the other hand, fluid-width sites adapt to the browser window and scale accordingly. These layouts, while popular, can be difficult to achieve when imagery is part of the structure. Fluidwidth sites are often good for content-rich sites but introduce a lot of nuances to the design process.

Different browsers render differently and fluid-width layouts make it difficult to completely address all types of user choices and preferences. Engadget (<http://engadget.com>), the popular gadget blog from AOL, employs a fluid-width layout. While certain elements are constricted to a minimum width, creating some fixed-width behavior at lower resolutions, when the browser is resized, many of the page elements are flexible.

Sidebars

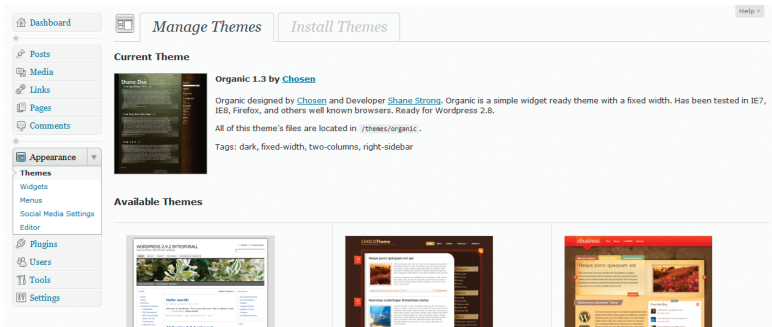
Most WordPress themes (and blogs in general) have sidebars. Depending on your site, you might want a single sidebar or more than one. You can even go with no sidebars. (Often, sites without sidebars have “normal” sidebar elements as part of their footers.)

Sidebars are often used to display ads, recent posts, recent comments, popular posts, or information from other social networks such as Flickr or Twitter. Because most themes are built to be able to use widgets, it is easy to configure or reconfigure them.

4.2 Finding and Implementing Free Themes

WordPress themes are the other side of WordPress extensibility. In Part II of this book, I took a long, hard look at plugins, hooks, widgets, and other aspects of extending WordPress with plugins. The other side of that equation is themes. As with plugins, WordPress has an entire theme infrastructure available for theme developers to use. The theme application programming interface (API) is comprised of an abundance of template tags and, as with plugins, themes can be installed with one click from inside the WordPress

Admin. Bloggers who want to install themes from within WordPress can do so by choosing **Appearance > Add New Themes**. This interface provides a rich set of filters based on colors, columns, layout type (fixed or fluid), feature set, or subject (a gooey kind of filter based around “intent of the theme”). In addition, you can search for themes by keyword. The filter interface is simply an interface that inserts the commonly used keywords associated with the filter name, such as “two-columns.”



Managing your themes is also easy with WordPress

The WordPress theme installer uses the WordPress Themes Directory located at [http://wordpress.org/ extend/themes/](http://wordpress.org/extend/themes/). As an example, you'll find a dark-colored theme with a fluid (also known as flexible) layout and threaded comment support.

1. Choose the **Appearance > Add New Theme** page in WordPress Admin, select the **Dark** check box in the **Colors** group, select the **Flexible** check box in the **Width** group, and select the **Threaded Comments** check box in the **Features** group.
2. Click **Find Themes** to perform a search of the available themes with those options in the WordPress Themes Directory.
3. Review the available options and click **Install** next to your desired theme. You can click **Preview** to see a sampling of what the theme will look like and click **Details** to get version and rating information about the theme. For this example, click the **Install** link under the **Charcoal** theme.
4. Click the **Install** button in the preview box. You will need to have your FTP (or if available, SFTP) connection information to enable WordPress to perform the install.
5. To activate the theme after a successful installation, click the **Activate** link.

You can also activate the theme from the [Appearance > Themes](#) page.

The WordPress Admin interface for installing themes. Once you have installed the theme (and optionally activated it), you can make theme modifications with CSS or adjust the code as needed. Once you activate it, you can customize it further with widgets.

WordPress with a theme that is configured to use widget, but with no widgets configured, has a default layout for the sidebar. This is coded according to the wishes of the theme designer but disappears once any widget is configured for use.

If you don't want to use the WordPress Admin to install a new theme, or if you want to use a theme not currently listed in the WordPress repository (there are still many themes that have not been added to the repository, are not necessarily under the General Public License (GPL), or are custom designed and made), you can manually upload your theme. To do this, follow these steps:

1. Download the theme zip file from the [WordPress Themes Directory](#) or third-party download site.
2. Extract the zip file. This creates a folder with all the theme files inside it.
3. Upload the theme via FTP or SFTP to the `/wp-content/themes` folder. For example, if the theme folder is called `cool-theme/`, you would upload the folder and end up with `/wp-content/themes/cool-theme/`.
4. From the WordPress Admin, choose [Appearance > Themes](#).
5. To Preview the theme before activating it, click the [Preview](#) link below your new theme. To activate the theme without previewing it first, skip to Step 7.
6. To activate the theme from the Preview window, click the [Activate](#) link in the upper right corner.
7. To activate the theme without previewing it first, click the [Activate](#) link under the theme listing.

4.3 Finding and implementing premium themes

Theme designers in the WordPress community are split down the GPL line. From the Automattic side of the house (the main proprietor of WordPress), there is no endorsement of premium themes (whether by explicit blessing or by implicit inclusion in the Themes Directory). In fact, this side of the argument has been historically hostile to the use of premium themes.

This position is based largely on some historical abuses by premium theme developers that carry licenses requiring inclusion of links back to the

developer or sponsor site, or even by inclusion of hidden keyword stuffing routines intended to generate traffic for some other source. Fortunately, this is not as common as it used to be when premium themes first arrived on the scene.

However, despite this abuse by some premium theme developers, many popular premium themes out there don't carry the same burden. These themes, such as the very popular and extensible Thesis theme from DIYThemes, provide the blogger with a multitude of configuration options, theme hooks, and WordPress Admin configuration interfaces.

Premium themes are different than free themes in one key area — you pay for them. These premium themes can cost as little as \$70 or as much as several hundred dollars. Some premium themes, like the ones available from WooThemes (<http://woothemes.com>) have licenses that are compatible with WordPress' license and give you as much freedom as you have with WordPress itself. Others, such as the Thesis theme, have proprietary licenses that restrict your use of the theme and may also require that you keep a link to the creator of the theme displayed on your site.

The Thesis theme is one premium theme that has so many built-in hooks and functionality that it has turned into somewhat of a theme framework. Premium themes are often a good compromise between using a common free theme available from the WordPress Themes Directory or other source, and having a completely unique custom designed theme that could cost thousands of dollars.

The screenshot displays the WordPress 'Manage Themes' screen. On the left is a sidebar with navigation links: Dashboard, Posts, Media, Links, Pages, Comments, and Appearance. The 'Appearance' menu is expanded, showing 'Themes', 'Widgets', 'Menus', and 'Social Media Settings'. The main content area has a header with 'Manage Themes' and 'Install Themes' tabs. Below this is a search bar with the text 'Search for themes by keyword, author, or tag.' and a 'Search' button. A 'Feature Filter' section is expanded, showing various theme features with checkboxes: Colors (Black, Orange, Silver, Dark, Blue, Pink, Tan, Light, Brown, Purple, White, Green, Red, Yellow), Columns (One Column, Two Columns, Three Columns, Four Columns, Left Sidebar, Right Sidebar), Width (Fixed Width, Flexible Width), and Features (Custom Colors, Custom Header, custom-background, custom-rr, editor-style, Theme Options, Threaded Comments, Sticky P). The 'Search' button is also visible in the top right of the main area.

Search for themes to suit your preferences using WordPress' search

4.4 Adding JavaScript and CSS to themes

As the internet evolves and more emphasis is placed on interactivity and user experience, the need for JavaScript increases. Using JavaScript has evolved from being painfully difficult, requiring browser detection and massive duplication of code to achieve a consistent user experience, to relatively simple, with implementations that work across all browsers. What once took developers 50 lines of code or more to achieve can now be done in just a few lines and is more reliable. JavaScript is a client-side language, meaning it is not executed on a server as PHP is. Instead, it relies on Web browsers to interpret it and handle it accordingly. The benefit is a simplistic concept of distributed computing, where the weight of processing and execution is not handled in a central spot, but distributed across all the computers accessing the Web page or Web application. In recent years, JavaScript has also been a cornerstone of highly interactive Web applications.

With the advent of Ajax (Asynchronous JavaScript and XML), Web applications gained the capability to use a hybrid model, where most of the user experience happens on the front end while background communication with servers handles most of the processing. This hybrid model provides a much more robust way to gain all the distribution and user experience of JavaScript with all the security and dynamic database communication of a server side system, such as PHP and MySQL. At the same time, user experience and themes require Cascading Style Sheets (CSS). CSS, like JavaScript, is a client-side technology. It is handled and parsed directly by the browser. There are a few versions of CSS, with varying levels of support from the various browsers and browser versions.

4.5 Examining the jQuery and Prototype Frameworks

WordPress makes it easy for developers to use the most common JavaScript frameworks available, jQuery and Prototype, and the Prototype-dependent framework, script.aculo.us. Most JavaScript in WordPress Admin is written in jQuery because it is a lightweight, but very powerful, library. However, script.aculo.us/Prototype provides richer animations and can be a more familiar development environment for some developers.

jQuery

jQuery, documented at www.jquery.com, powers most of the JavaScript in the WordPress Admin. However, you can also use it in themes by enqueueing (which means “adding a new item to the queue”) the script in a function or plugins.

The jQuery framework operates on the principle of chaining, or connecting expressions end to end. It usually uses a dollar sign (\$) to begin an expression. jQuery developers will often replace the \$ notation with the word “jQuery” because Prototype developers are used to the identical notation. In WordPress, for compatibility reasons, jQuery is used in this “no conflict” mode, which uses jQuery instead.

Technically, the jQuery() function in jQuery literally means “an element designated by a CSS string.” While jQuery often uses complex selectors, it always references a CSS selector. This differs from Prototype, which uses Document Object Model (DOM) IDs, discussed later in this chapter, where \$ indicates “element with ID of.” The nuance between these two paradigms is slight, but is extremely important to understand in development. To use jQuery in a theme, you must have a ready event. The ready event is a statement that ensures that the entire page, or DOM as it’s known in JavaScript development, has been loaded. It is expressed as jQuery(document).ready() and wraps around all jQuery expressions.

Wrapping all expressions in a ready event

```
<script type="text/javascript">
jQuery(document).ready(function() {
// Insert all jQuery expressions here
});
</script>
```

In a practical sense, you can use jQuery for almost anything. For example, assume that you implement a search form in a hidden layer. The way this form is exposed is by clicking on a link.

Basic HTML structure for a div that contains a search form and a link

```
<a id="search_link" href="#">Search</a>
<div id="example_search_form">
<?php get_search_form() ?>
</div>
```

Using a function to add CSS to the theme

```
function additional_css()
{
?>
<style type="text/css">
#example_search_form { display:none; }
</style>
```

```
<?php
}
add_action('wp_head','additional_css');
```

Using jQuery to create a click event to display the form

```
function additional_js()
{
    wp_enqueue_script('jquery');
    wp_print_scripts();
    ?>
<script type="text/javascript">
jQuery(document).ready(function() {
jQuery('a#search_link').click(function() {
jQuery('#example_search_form').show("slow");
});
});
</script>
<?php
}
add_action('wp_head','additional_js');
```

Clearly, this is a simplistic example. You can use jQuery to provide many different interactive possibilities to a Web site and, for the WordPress Admin, it is the preferred JavaScript framework to use because it is already in use throughout the backend.

Prototype

The second JavaScript framework that is bundled and available to WordPress developers by default is Prototype (documented at www.prototypejs.org/). Like jQuery, Prototype uses the \$ function throughout but unlike jQuery, it does not provide a way for a “no conflict” mode. Prototype is used by the WordPress core in only three parts of the WordPress Admin. Primarily, it is used in conjunction with TinyMCE (the rich text editor provided by WordPress for writing posts), the flash-based media uploader, and the Blogger import script.

The media uploader in WordPress is based on Flash and Prototype. However, Prototype was used more in the past, but many plugins still rely on the framework, so it is still part of a bundled JavaScript framework, and some of the core features of WordPress make use of it.

Using Prototype, the hidden search box can be exposed by clicking the link with ID #search_link

```
function wpb_enqueue_prototype()
{
    wp_enqueue_script('prototype');
}
function wpb_additional_js()
{
    wp_enqueue_script('prototype');
    wp_print_scripts();
?>
<script type="text/javascript">
var HiddenSearchShower = Class.create({
  initialize: function(field) {
    Event.observe(document, 'click', this.showSearch );
  },
  showSearch: function(event) {
    $('#example_search_form').setStyle({display:'block'});
  }
});
var SearchForm = new HiddenSearchShower("#search_link");
</script>
<?php
}
add_action('wp_head', 'wpb_additional_js');
add_action('wp_enqueue_scripts', 'wpb_enqueue_prototype');
```

Comparing the frameworks

Both jQuery and Prototype are bundled in WordPress. Developers familiar with other JavaScript frameworks (such as Moo Tools, Dojo, or Spry) are free to add these libraries via a plugin (using `wp_register_script()` or `wp_enqueue_script()`), but developers who focus in WordPress tend to use the bundled packages. Both frameworks offer benefits and drawbacks. For example, jQuery provides a light framework that mixes in some effects and has a robust plugin architecture to extends its core use, but it may not be as robust as a scripting language for user-interface/webapp-style blogs.

On the other hand, Prototype is a rich object-oriented JavaScript framework that will be familiar to developers from the Ruby on Rails world,

but lacks the included effects that jQuery offers. For that, developers will need `script.aculo.us`, a library that is discussed later in this chapter. Both frameworks have benefits, but the libraries of code built around them, in the form of plugins or extensions, is what really makes them useful in WordPress.

4.6 Leveraging WordPress' JavaScript libraries

JavaScript frameworks bundled in WordPress are powerful pieces of software that allow for quick and clean JavaScript development. With these libraries, developers can radically alter the way a theme behaves or how a user interacts with the WordPress Admin. The preferred library in WordPress development is jQuery, but some developers are just more familiar with Prototype, especially if they have come from a Ruby on Rails background. Both libraries can be used in conjunction with each other.

In the next section, I'll introduce you to the JavaScript libraries (extensions to the JavaScript frameworks) that are bundled with WordPress. This should not prevent developers from including other frameworks manually if they prefer to do so.

Prototype: `script.aculo.us`

Built on Prototype, `script.aculo.us` (<http://script.aculo.us/>) is a series of libraries that provides effects, animations, and other user interface enhancements. It is not used anywhere in the WordPress core, but is provided as a legacy library for plugins that need it. The `script.aculo.us` library provides plenty of effects that can be useful in plugin development. Some include the capability to make page elements semi-transparent (opacity) in a cross-browser fashion, moving elements on the screen, and Ajax data retrieval.

The main downfall to `script.aculo.us` is its weight (it weighs in at a hefty 632K), so it is not recommended that plugins load the entire library. You can choose to load the entire suite or just the portions needed, but make sure you load all the dependent libraries as well.

Prototype: TinyMCE

Many rich text editors are available on the web. TinyMCE was adopted into the WordPress core because it is lightweight, is entirely General Public License (GPL) – compatible, and the team behind TinyMCE was willing to work closely with WordPress to develop a product that could be distributed to millions of blogs.

TinyMCE, created by Moxiecode (<http://tinymce.moxiecode.com/>), can turn a form into a rich editing experience where bloggers can write stories in an interface that gives them formatting controls similar to what they are familiar with in a word processing program. Outside of the WordPress Admin, another common use for TinyMCE is with the comment box on a theme. This approach enables readers to leave rich comments without having to use HTML. The TinyMCEComments plugin (<http://wordpress.org/extend/plugins/tinymcecomments/>) does just that. If a theme has a comment form hook in the comments template, this plugin hooks into the form, hijacks the `<textarea>` tag, and replaces it with the TinyMCE editor.

TinyMCE provides the rich text editing experience in the WordPress Admin. The plugin, TinyMCEComments, replaces the standard comment `<textarea>` with an instance of the TinyMCE editor.

jQuery: Autosave

Autosave is a WordPress-specific jQuery plugin that saves posts while the blogger writes. It uses the “autosave” handle and offers limited additional benefit outside of saving posts.

jQuery: hoverIntent

This plugin is available to WordPress with the handle “hoverIntent”. The hoverIntent plugin works with mouse movement to determine if a user is intending to hover over an element. If it determines that mouse speed has slowed, for example, it might fire the mouseover DOM event.

jQuery: Farbtastic

Farbtastic is a popular jQuery plugin that enables plugin developers to include an interactive color picker. It is referred to with the handle “farbtastic” and you can find documentation on it at <http://acko.net/dev/farbtastic>. Currently, it is only used in the WordPress core to allow bloggers to customize the color of the header in the default theme; however, it is bundled to make the library available to other plugins.

The Farbtastic jQuery plugin provides a color picker that can be used in WordPress. The default theme uses it to enable bloggers to pick their header color.

jQuery: jQueryUI

The most important set of jQuery plugins to WordPress are the jQueryUI

plugins. This suite of plugins provides the drag-and-drop widgets interface, and the capability to reorganize and order the WordPress Dashboard and Write Post/Page screens, as well as manage all the user interface effects in the WordPress Admin navigation.

Of course, the effects used here are only a fraction of the possibilities available to plugin developers. At a very minimum, jQueryUI can be loaded using the `jquery-ui-core` handle. Other plugins can be loaded using similar handles, but all must have both `jquery` and `jqueryui-core` as dependencies when enqueued.

4.7 Examining WordPress Plugins that Use JavaScript Effects

As the Web world gets more interactive, creative plugin developers continue to add more interactivity to blogs. Much of this interactivity is as a result of the bundled JavaScript frameworks and libraries. These plugins are free to use and may provide inspiration for what is possible when the frameworks are used. They can give you a sense of how to develop JavaScript code for use in plugins if you are just learning.

jQuery Reply to Comment

With WordPress 2.7, the commenting application programming interface (API) was rewritten to support threaded and paged comments. When Threaded comments are enabled, Reply links are provided with each comment. Users can simply click the Reply link to create a threaded conversation around an existing conversation. The jQuery Reply to Comment plugin (<http://wordpress.org/extend/plugins/jqueryreply-to-comment/>) takes this concept one step farther by allowing commenters to quote a comment with their replies.

Social Bookmarks

The Social Bookmarks plugin (<http://wordpress.org/extend/plugins/socialbookmarks/>) is like the smattering of their social tools integration plugins. It adds a selection of social network icons to the end of posts, enabling readers to share your articles on those networks. It uses script.aculo.us for all JavaScript functionality.

WP Conditional Digg This Badge

The WP Conditional Digg This Badge plugin (<http://wordpress.org/extend/plugins/conditional-digg-this-badge/>) uses Prototype to access the Digg.com

API to determine how many “Diggs” an article has. Using conditional logic based on blogger-defined parameters (such as how many Diggs are required), it displays an interactive Digg This badge in a post.

WPTouch iPhone Theme

Given the popularity of the iPhone these days, iPhone-specific versions of a blog are in demand. The WPhone iPhone Theme plugin (<http://wordpress.org/extend/plugins/wptouch/>) uses jQuery to render a version of a WordPress blog that is formatted for the iPhone. The WPTouch iPhone Theme plugin renders a WordPress blog in an iPhone format. It uses jQuery.

4.8 Looking at theme styles

CSS order is not as important as JavaScript library order, but you do have to remember the “cascading” nature of Cascading Style Sheets. For example, with JavaScript, you encounter conflicts that have the potential of breaking a page if you load script.aculo.us before you load Prototype. This is why the `wp_enqueue_script()` function exists. Dependencies can be determined and scripts injected into the blog in the appropriate order, avoiding potentially site crippling side effects.

4.9 Queuing styles

While the `wp_enqueue_style()` function effectively does the same thing, CSS conflicts generally won’t cripple a site. However, the nature of CSS is to “cascade.” Styles cascade from a top-level hierarchy, and can be overridden later in a page execution. To this end, conflict is avoided using a similar syntax to `wp_enqueue_script()`. The first argument passed to `wp_enqueue_style()` must be a unique handler. The second argument should be the fully qualified URL to the stylesheet, while the third argument, which is completely optional, should be an array of dependencies (using the dependency handler name).

```
{
wp_enqueue_style('my-styles', WP_CONTENT_DIR . '/themes/
my-theme/style.
css');
wp_enqueue_style('ie-my-styles', WP_CONTENT_DIR . '/
themes/my-theme/
ie.css', array('my-styles'));
wp_print_styles();
```

```
}  
add_action('wp_enqueue_scripts','wpb_additional_css');
```

Using JavaScript to style elements

Sometimes it is necessary to use JavaScript to adjust styles on the fly. This is because some kind of event (hover, click, and so on) has occurred on a page after all the CSS files have been loaded and applied when the page initially loads. Fortunately, both the jQuery and Prototype frameworks discussed earlier in this chapter have ways to apply styling to an element after the page loads. For example, maybe you want to change the background color of a post when you hover over it. Using the default WordPress theme, you can write a function that adds some jQuery inside the <head> tag.

Using jQuery to change the hover background CSS of a post

```
<script type="text/JavaScript">  
jQuery.noConflict();  
jQuery(document).ready(function() {  
jQuery('.entry').hover(  
function() {  
jQuery(this).css('background', '#eee');  
},  
function() {  
jQuery(this).css('background', '#fff');  
}  
});  
</script>
```

Using jQuery, a plugin can alter the defined style of an element. In this case, the background color of a post changes to a light grey when you hover over it. and styles can be deregistered if they're implemented with the style queue.

JavaScript can be used to alter CSS, creating a dynamic user experience. **d**

IMPORTANT

We've put the last chapter of this Fast Track on the Mindware DVD. Please read it from there. Write to editor@thinkdigit.com and tell us whether you prefer reading on screen or on paper. The way in which you get Fast Tracks in the future depends on this, so make sure to write in!

5 WORDPRESS PLUGINS

WordPress Plugins are composed of PHP scripts that add additional functionality to your blog. They offer new additions to your blog that either enhance features that were already available or add otherwise unavailable new features to your site.

Plugins extend the functionality of WordPress. The majority of WordPress users don't require Plugins, or only require a few, such as Plugins dealing with comment spam or customised post listings. Other users enjoy the varied options Plugins provide such as frequently updated weather reports, post word counts, rating systems, and more. Since the choice in Plugins is vast, and chosen based upon the needs of the user, they are not incorporated into the core of WordPress.

5.1 Installing Plugins

WordPress offers simple and easy ways of adding Plugins to your blog. From the Administration Panels, click on the Plugin tab. Once you have uploaded a plugin to your WordPress plugin directory, activate it from the Plugins Management page, and sit back and watch your plugin work.

There are two ways to do that:

- WordPress built-in installer – search, install and activate completely in the Add New menu.
- Traditional manual installation - get and expand zip files, upload to your sever, and activate.

5.1.1 Manual Plugin Installation

To install a plugin that is not in the plugin browser, you should follow the given steps. Don't forget to backup just in case.

1. Upload the plugin to the wp-content/plugins folder in your WordPress directory online.
2. Make any changes to templates or files as required by the Plugin instructions including adding Plugin template tags.
3. Activate the Plugin:
 - a. Access the Plugin Panel in your Administration Panels
 - b. Scroll down through the list of Plugins to find the newly installed Plugin (if not visible, start from the beginning to check to see if you followed the instructions properly and uploaded the file correctly).

c. Click on the Activate link to turn the Plugin on.

5.1.2 Hiding plugins when deactivated

Some plugins feature tags inside of the template files. If the plugin is not activated, it will "break" the Theme and it may report errors or fail to load. It is therefore imperative to prevent the plugin from being detected in case it is turned off.

To detect if a plugin is installed, you can use a simple `function_exists()` check. The `if (function_exists())` checks for the plugin, and if it exists, it will use it. If it returns `FALSE` or "not found", it will ignore the plugin tag and continue loading the page.

```
<?php
if (function_exists('FUNCTION_NAME')) {
    FUNCTION_NAME();
}
?>
```

This example plugin uses a function called `jal_get_shoutbox()` to print out its contents.

```
<?php
if (function_exists('jal_get_shoutbox')) {
    jal_get_shoutbox();
}
?>
```

5.2 Troubleshooting plugins

If you are experiencing problems with a plugin you installed or one that stopped working after upgrading, the following are the steps you need to take to troubleshoot the plugin:

1. Check that any plugin tags or usage within your template files are correct, spelled right, and placed in the appropriate place, i.e., within the WordPress Loop or outside of it.
2. Check that you uploaded the file to the plugins folder under wp-content.
3. Check that the plugin has been activated in your Plugin Panel of your Administration Panels.
4. Deactivate and re-activate the plugin to see if this makes it work.

If the problem persists and you cannot seem to solve it, just check for similar plugins that do the same work. You will find plenty!

5.3 Plugin management

Plugins are managed from the Plugins panel in the Administration panel of your WordPress site. All plugins listed on this screen are found in your `wp-content/plugins` directory. Each plugin has a description of what it does, an author and web site to refer to, and a version number.

5.3.1 Activation and deactivation

If your plugin requires changes to the WordPress code or your template files, you will need to enact or reverse those changes each time you activate or deactivate your plugin. Failing to do this will likely result in errors. There is also a link provided below the list of Plugins that allows you to activate All Plugins.

5.3.2 Uninstalling plugins

While plugins are very useful, some plugins just don't meet your needs, or you've stopped using them for some reason, or they just don't work. After deactivating the Plugin stops the Plugin's behavior, they tend to pile up in your Plugins panel, making your plugin list long and cumbersome to scroll through. If left there long enough, they might become obsolete with the newer WordPress versions and cause problems if you decide to use them in the future.

To remove a plugin, make sure the plugin is deactivated from the Plugins panel. Go to your web site's `wp-content/plugins` folder (usually with an FTP program) and look for the file name of the plugin you want to remove. Select the file name and delete it.

5.3.3 Things to know before you install

There are a few things you need to know before you install WordPress plugins. Plugins require downloading and uploading to install. You must be familiar with how to download and upload files and how to use FTP, if required. Plugins may require modifying WordPress files and templates. Familiarity with PHP, HTML, CSS, and CHMOD may be necessary.

Record modifications: If you make any changes to the WordPress files or templates, make a note of it in the code by using comments before and after the changes, and in a text file saved to your site and on your desktop to remind you of any additions or modifications to your default files or templates. This will help you repeat these in the future if there are any problems with your site.

Take frequent backups: Some plugins work independently of your blog's content and files, just adding a little something special to the page. Others require modification of content and files, including changes to your database. Before installing any plugin that will make dramatic changes, backup your database and files.

5.4 Developing your own plugins

Once you start using WordPress Plugins, you sometimes wonder how you ever got along without them. If you have knowledge of PHP, you can develop your own plugins, but detailed instructions are beyond the scope of this book.

5.5 Most popular Wordpress plugins

1. All in One SEO Pack - <http://wordpress.org/extend/plugins/all-in-one-seo-pack/>

It automatically optimises your WordPress blog for search engines: page titles, meta tags, keywords, and descriptions. This plugin allows you to configure them for either your entire blog or on a post by post basis.

2. Google Analytics for WordPress - <http://wordpress.org/extend/plugins/google-analytics-for-wordpress/>

The Google Analytics for WordPress plugin allows you to track your blog easily. It adds the asynchronous tracking code for Google Analytics and automatically tracks and segments all outbound links from within posts, comment author links, links within comments, blogroll links and downloads. It'll add custom variables while tracking to show you page views per author, post type, category or publication year. It'll also allow you to track AdSense clicks, add extra search engines, and even ignore certain user levels.

3. Contact Form 7 - <http://wordpress.org/extend/plugins/contact-form-7/>

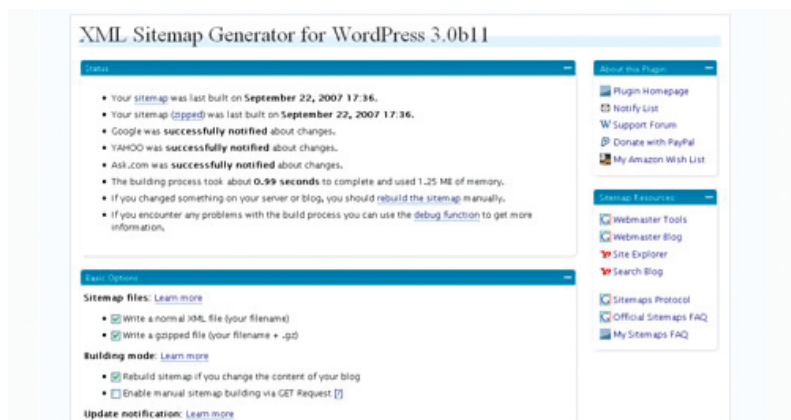
Contact Form 7 can manage multiple contact forms, plus you can customise the form and the mail contents flexibly with simple markup. The form supports Ajax-powered submitting, captcha, and Akismet spam filtering.

4. Akismet - <http://wordpress.org/extend/plugins/akismet/>

Akismet checks your comments against the Akismet web service to see if they look like spam or not and lets you review the spam it catches under your blog's "Comments" admin screen.

5. Google XML Sitemaps - <http://wordpress.org/extend/plugins/google-sitemap-generator/>

This plugin generates a special XML sitemap that will help search engines such as Google, Bing, Yahoo and Ask.com to better index your blog. With such a sitemap, it's much easier for the crawlers to see the complete structure of your site and retrieve it more efficiently. The plugin supports all kinds of WordPress generated pages as well as custom URLs. Additionally, it notifies all major search engines every time you create a post about the new content.



The Sitemap generator will create a sitemap for your web site

6. WPTouch iPhone Theme- <http://wordpress.org/extend/plugins/wptouch/>

WPTouch automatically transforms your WordPress blog into an iPhone application-style theme, complete with AJAX loading articles and effects, when viewed from an iPhone, iPod touch, Android, Opera Mini, Palm Pre and BlackBerry Storm mobile devices. The admin panel allows you to customise many aspects of its appearance, and deliver a fast, user-friendly and stylish version of your site to iPhone, iPod touch, Android, Palm Pre and BlackBerry Storm visitors, without modifying a single bit of code (or affecting) your regular desktop theme. The theme also includes the ability for your visitors to easily switch between the WPTouch view and your site's regular theme.

7. Fast and Secure Contact Form

<http://wordpress.org/extend/plugins/si-contact-form/>

This contact form lets your visitors send you a quick E-mail message. It

also blocks all common spammer tactics. Spam will no longer be a problem because of its CAPTCHA and Akismet support. Additionally, the plugin has a multi-form feature, optional extra fields, and an option to redirect visitors to any URL after the message is sent.


8. Post videos and photo galleries - <http://wordpress.org/extend/plugins/post-rich-videos-and-photos-galleries/>

This plugin lets you post your videos and photo galleries/flash slideshows easily and in seconds. It creates an automated, fully customisable image gallery, slideshows, video and music playlists anywhere within your WordPress site. You can then choose your videos, images and music and display skin, pages or posts with custom overlay text and a rotating thumbnail belts.

9. WP Super Cache - <http://wordpress.org/extend/plugins/wp-super-cache/>

This plugin for WordPress produces static HTML files from your dynamic WordPress blog. After a HTML file is generated, your web server will serve that file instead of processing the comparatively heavier and more expensive WordPress PHP scripts. The static HTML files will be served to the vast majority of your users, but because a user's details are displayed in the comment form after they leave a comment those requests are handled by PHP. Static files are served to:

1. Users who are not logged in.
2. Users who have not left a comment on your blog.
3. Or users who have not viewed a password protected post.

Ninety nine per cent of your visitors will be served static HTML files. Those users who don't see the static files will still benefit because they will see regular WP-Cache cached files and your server won't be as busy as before. This plugin will help your server cope with a front page appearance on digg.com or other social networking sites. 

IMPORTANT

We've put the last chapter of this Fast Track on the Mindware DVD. Please read it from there. Write to editor@thinkdigit.com and tell us whether you prefer reading on screen or on paper. The way in which you get Fast Tracks in the future depends on this, so make sure to write in!

6 WIDGETS

6.1 Making Your Content Unbreakable

There is one big drawback to using WordPress: the lack of custom content types/groups, an area where developers put restrictions on how clients insert content. This is fairly easy to do with some knowledge of custom fields, but can be a little complicated if your client is new to WordPress. Developers must create workarounds to keep the content clean, portable and relatively unbreakable.

1. More Fields - <http://wordpress.org/extend/plugins/more-fields/>

More Fields is a WordPress plugin that adds boxes to the Write/Edit page. These boxes contains input fields, so that additional (more) fields can be added to a post. For example, if you write about books, you can add a box where you can enter title and author, etc. The boxes can be placed either to the right or to the left on the Write/Edit page.

2. Creating Custom Content Type with Flutter - <http://flutter.freshout.us/>

Flutter is made precisely for CMS-making, allows you to create custom Write Panel (Posts, Pages, and Theme Options). This, basically, is a custom content type in which you can add your own fields. So, if a WordPress installation gives you two different content types (Post and Page), Flutter enables you to add more Write Panels to your likings. You can create any type of field (File Uploads, Audio, Video, Checkboxes, Dropdowns) to make as simple as possible for your client or yourself to enter content.

3. Post Templates - <http://wordpress.org/extend/plugins/post-template/>

If you find yourself writing posts and pages on a regular basis which have the same structure, then this plugin would be a great addition to your WordPress plugin arsenal. This could be very useful for clients who have no idea how to get started to write a new page or post. Written by Vincent Prat, Post Templates plugin helps you save time blogging by letting you save and recall templates for writing posts and pages.

4. WP-CMS Post Control

<http://wordpress.org/extend/plugins/wp-cms-post-control/>

This plugin gives you complete control over your write options. It not only

allows you to hide unwanted items like custom fields, trackbacks, revisions etc... but also gives you a whole lot more control over how WordPress deals with creating content! This helps you use WordPress more like a CMS, allowing you to totally customise what your authors see and use.

6.2 Managing Users and their Roles

1. User Access Manager

<http://wordpress.org/extend/plugins/user-access-manager/>

This plugin is useful if you need a member area or a private section at your blog or you want that other people can write at your blog but not everywhere. Features include:

- User groups
- Set separate access for readers and editors
- Set access by user groups
- Set access by post categories
- User-defined post/page title (if no access)
- User-defined post/page text (if no access)
- Optional login form (if no access)
- User-defined comment text (if no access)

2. Role Scoper

<http://wordpress.org/extend/plugins/role-scoper/>

Role Scoper is a comprehensive enrichment for access control in WordPress, giving you CMS-like control of permissions. Assign reading, editing or administration roles to users or groups on a page-specific, category-specific or other content-specific basis.

3. Role Manager

<http://www.im-web-gefunden.de/wordpress-plugins/role-manager/>

This Plugin allows you to define and manage multiple subscriber profiles – called Roles and their Capabilities. Also you can create new Roles and Capabilities.

4. Member Access - <http://wordpress.org/extend/plugins/member-access/>

Member Access allows a WordPress administrator to make individual posts and pages accessible only to logged-in members. Member Access allows global configuration so that all posts or pages can be viewable by everyone

(the default) or only by members, and it also allows each post and page to override the global setting.

6.3 Better & Smart Navigation

1. WordPress Navigation List

<http://wordpress.org/extend/plugins/wordpress-navigation-list-plugin-navt/>

The plugin gives you the ability to create unique site navigation from your pages, categories and users using a Drag 'n Drop Interface; arrange the items within a group in any arbitrary order. Navigation groups may be composed of any combination of pages, categories, Authors, (Editors, Contributors, Subscribers), internal/external links and list dividers. This plugin has a lot of other great features, you have to check it out.

2. Multi-level Navigation

<http://wordpress.org/extend/plugins/multi-level-navigation-plugin/>

Adds an SEO friendly, accessible dropdown/flyout/slider menu to your WordPress blog. The plugin generates the code necessary to create a Son of Suckerfish horizontal dropdown, vertical flyout or horizontal slider menu. The plugin produces W3C valid HTML and CSS and only requires Javascript to function for very old browsers such as IE 6. Check out the demo here. The options page allows you to choose from a range of options including:

- What content will be displayed in the menu
- Animation speed (how fast the dropdowns appear)
- Mouseover delay
- Hide delay
- Add a second menu

3. Yoast Breadcrumbs - <http://yoast.com/wordpress/breadcrumbs/>

Breadcrumbs are the links, usually above the title post, that look like Home > Articles > WordPress SEO. They are good for two things: a) They allow your users to easily navigate your site. b) They allow search engines to determine the structure of your site more easily.

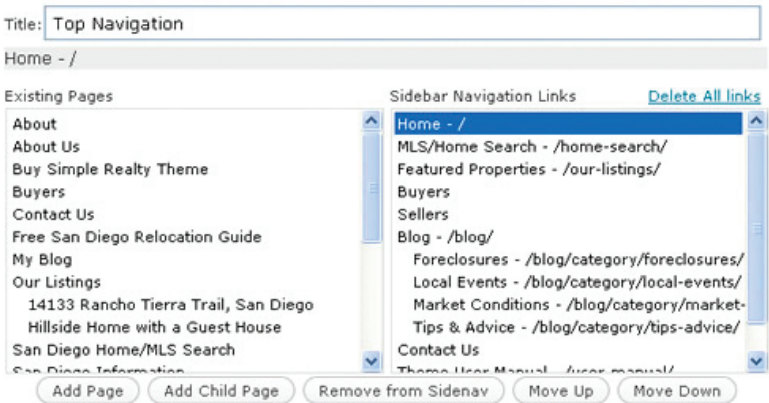
These breadcrumbs should link back to the homepage, and the category the post is in. This plugin allows you to add breadcrumbs to your theme.

4. Simple Sidebar Navigation

<http://wordpress.org/extend/plugins/simple-sidebar-navigation/>

This plugin adds a widget into the Widgets dashboard that allows in a simple way, without writing any PHP code on your part, to create custom navigation in any sidebar pre-defined by your WP theme. Features include:

- Flat and multi-level navigation hierarchy for existing pages and custom links.
- Very flexible conditional appearance options.
- Out-of-the-box Suckerfish support.
- Option to add navigation links with custom title, url and target attribute.
- Unlimited number of navigation widgets.
- Optional setting includes blog posts into the navigation selection list (only pages are available by default).



Sidebar navigation supports custom drop-down menus CSS

6.4 Having a multilingual site

Writing multilingual content is already hard enough, so why go extra mile without making sure you already have a good plan to manage translation on your blog. Check out the plugins below that will help you have a fully featured multilingual CMS.

1. WPML Multilingual CMS

<http://wordpress.org/extend/plugins/sitepress-multilingual-cms/>

Turns WordPress sites into a fully featured multilingual content management system (CMS). WPML helps it go the extra mile. Features include: 1) Multilingual content support based on Drupal i18n architecture. 2) CMS navigation allows

adding drop down menus, breadcrumbs trail and sidebar navigation (all widgetized). 3) Creates internal Sticky Links so that they never break.

2. xLanguage - <http://wordpress.org/extend/plugins/xlanguage>

xLanguage is a full featured plugin allows you to blog in different language, and allows user to select which version to read. It works for blog post, page, tags, categories. The user language preferences will also select the right theme and plugins MO files.

3. qTranslate - <http://wordpress.org/extend/plugins/qtranslate>

qTranslate makes creation of multilingual content as easy as working with a single language. Here are some features:

- qTranslate Services – Professional human and automated machine translation with two clicks
- One-Click-Switching between the languages – Change the language as easy as switching between Visual and HTML
- Language customizations without changing the .mo files – Use Quick-Tags instead for easy localization
- Multilingual dates out of the box – Translates dates and time for you
- Comes with a lot of languages already builtin! – English, German, Simplified Chinese and a lot of others

6.5 Easily Manage Categories, Pages and Posts

1. Idealien Category Enhancements

<http://wordpress.org/extend/plugins/idealien-category-enhancements/>

Manage category templates as easily as you manage page templates. Select which apply through the post > categories menu using file names that make sense, not category ID numbers. Now any views of the category (or sub-categories) will render the according to the category template selected. Posts in a given category can also use a category-based template.

2. pageMash > Page Management

<http://wordpress.org/extend/plugins/pagemash/>

Customise the order your pages are listed in and manage the parent structure with this simple AJAX drag-and-drop administrative interface with an option to toggle the page to be hidden from output. Great tool to quickly re-arrange your page menus.

3. Exclude Pages

<http://wordpress.org/extend/plugins/exclude-pages/>

This plugin adds a checkbox, “include this page in menus”, which is checked by default. If you un-check it, the page will not appear in any listings of pages (which includes, and is usually limited to, your page navigation menus).

4. WP No Category Base

<http://wordpress.org/extend/plugins/wp-no-category-base/>

As the name suggests this plugin will completely remove the mandatory ‘Category Base’ from your category permalinks (e.g. myblog.com/category/my-category/ to myblog.com/my-category/).

6.6 SEO

WordPress by default is pretty decent at letting search engines see what’s going on. But there are a whole bunch of plugins that can be used to make your blog rank better.

1. Redirection - <http://wordpress.org/extend/plugins/redirection/>

Redirection is a WordPress plugin to manage 301 redirections, keep track of 404 errors, and generally tidy up any loose ends your site may have. This is particularly useful if you are migrating pages from an old website, or are changing the directory of your WordPress installation.

2. Permalinks Moved Permanently

<http://wordpress.org/extend/plugins/permalinks-moved-permanently/>

If you just migrated your WordPress blog from one permalink structure to another, and you don’t want to lose Pagerank or traffic that accesses your blog through the old permalinks, this is for you.

3. SEO Smart Links - <http://wordpress.org/extend/plugins/seo-automatic-links/>

SEO Smart Links provides automatic SEO benefits for your site in addition to custom keyword lists, nofollow and much more. SEO Smart Links can automatically link keywords and phrases in your posts and comments with corresponding posts, pages, categories and tags on your blog.

4. Platinum SEO Pack - <http://wordpress.org/extend/plugins/platinum-seo-pack/>

This is an improvised version of the AllinOne SEO plugin. It does everything that AllinOne was doing and also some extra cool features. It takes care of

pretty much everything, a bit more advanced than a basic SEO plugin but easy to use as well.

6.7 Highly customisable Widgets

1. Widgets Reloaded - <http://wordpress.org/extend/plugins/widgets-reloaded/>

Widgets Reloaded replaces many of the default widgets with versions that allow much more control. Widgets come with highly customizable control panels. Each widget can also be used any number of times.

2. Flexi Pages Widget - <http://wordpress.org/extend/plugins/flexi-pages-widget/>

Flexi Pages Widget is a highly configurable WordPress sidebar widget to list pages and sub-pages. Can be used as an alternative to the default 'Pages' widget. Features include:

- Option to display sub-pages only in parent page and related pages.
- Option to select and exclude certain pages from getting displayed in the list.
- Option to include a link to the home page.
- Other options include title, sort column/order, hierarchical/flat format, show date.
- Multiple instances of the widget. Unlimited number of instances of the widget can be added to the sidebar.

3. SidePosts Widget - <http://wordpress.org/extend/plugins/sideposts/>

With this widget you select the category you want, and all entries with this category, will be shown on the sidebar instead the main blog. You will have then a small blog on the sidebar for those special entries. For each entry, you have the link to the post page. You can select the number to post to show and if must show only the post excerpt or the full post content (Also excerpt with thumbnails can be shown).

4. Query Posts Widget

<http://justintadlock.com/archives/2009/03/15/query-posts-widget-wordpress-plugin>

This widget has everything you'll ever need to show posts on your site without touching code. It's like having a WordPress developer ready to do your bidding all wrapped up in a widget. All you need is a theme that's widgetised.

It lets you displays posts by nearly any criteria you can imagine using the `query_posts()` function, except you don't need to know anything about PHP to use it. You can do all kinds of neat things: Posts by tag, Posts by

category, Posts by author, Posts by time/date, Posts by custom field key and/or value, Choose any number, Show pages, Show the full post, excerpt, or even order them in a list, and much more...

6.8 Must check plugins

1. Search Everything - <http://wordpress.org/extend/plugins/search-everything/>

Search Everything increases the ability of the default WordPress Search, options included: Search Every Page, Search Every Tag, Search Every Category, Search non-password protected pages only, Search Every Comment, Search Every Custom Field, Exclude Posts from search, Exclude Categories from search and more.

2. Subscribe2 - <http://wordpress.org/extend/plugins/subscribe2/>

Subscribe2 provides a comprehensive subscription management and email notification system for WordPress blogs that sends email notifications to a list of subscribers when you publish new content to your blog. Email Notifications can be sent on a per-post basis or periodically in a Digest email.

3. Exec-PHP - <http://wordpress.org/extend/plugins/exec-php/>

The Exec-PHP plugin executes PHP code in posts, pages and text widgets. Features include

- Executes PHP code in the excerpt and the content portion of your posts and pages
- Configurable execution of PHP code in text widgets (for WordPress 2.2 or higher)
- Write PHP code in familiar syntax, eg.

4. Viper's Video Quicktags

<http://wordpress.org/extend/plugins/vipers-video-quicktags/>

Tired of copying and pasting the embed HTML from sites like YouTube? Then this plugin is for you. Simply click one of the new buttons that this plugin adds to the write screen (rich editor included) and then paste the URL that the video is located at into the prompt box — easy as that. You can fully configure how the videos are displayed (width, height, colors, alignment on the page) and much more.

6.9 Best WordPress CMS Plugins

For moderately sized sites (including simple e-Commerce sites), WordPress

does a pretty good job as a CMS, making it easy to maintain your site, and update your content. Of course, it does this best with the help of a good theme, and some great plugins. The strength of WordPress is the community of developers who have already done almost anything you can think of with it. Here are the best plugins we've run across, the ones we install for nearly all of our client's sites.

1. Cforms II - <http://deliciousdays.com/cforms-plugin>

This is far and away the best contact form plugin there is. The reason? configurability. You can easily build literally anything with it. We've built out employment application forms, wedding checklists, and more for clients. It also drives our own quote form. There's also built in spam protection, via question and answer or captcha.

2. WP e-Commerce - <http://www.instinct.co.nz/e-commerce/>

Unlike cforms, this plugin has no competition. It provides an easy way to manage storefront and shopping cart as an integral part of your WordPress installation. It includes the ability to sell digital downloads as well, which is great for photographers, musicians, and authors. By default it includes support for PayPal, although the \$29.99 fee for the authorize.net module will pay for itself in lower fees for most serious e-Commerce sites.

3. Google Sitemap Generator

<http://www.arnebrachhold.de/redir/sitemap-home/>

The biggest benefit of using WordPress is the manual labor you save because the software already knows where all of your content is. This Plugin submits a comprehensive index of your site to Google, Yahoo, MSN Live, and Ask.com every time you update your site. It's a huge boost to your site's SEO.

4. Event Calendar 3 - <http://wpcal.firetree.net/>

There are a ton of event plugins out there. This is the best & most integrated one we've used. It provides an iCal feed, in addition to the standard RSS feed. Since the events are created as Blog posts, it's easy to integrate them into your site.

5. Page Links To - <http://txfx.net/code/wordpress/page-links-to/>


By far the simplest plugin on this list, This plugin allows you to create "pages" that link to something else. It's the easiest way to add links to your main menu, without mucking around in code. It's good for adding a link to an external

photography portfolio, for example. You could also use it to link to sections of a page, rather than an entirely separate page, which could be useful for linking to sections of a restaurant menu, or other small sections of a larger page.

6. TinyMCE Advanced – <http://www.laptoptips.ca/projects/tinymce-advanced/>

The plugin does a great job of making tables work well with WordPress posts.

7. XSPF_Player – <http://www.boriel.com/>

As much as we're not fans of auto-playing music on web sites, This MP3 player makes it easy to manage playlists, cover art, and artist links. It's a great way for music venues to highlight upcoming shows. Used in conjunction with Event Calendar, you can make the track link lead to the post for their upcoming show, all of which is easily manageable from your wordpress dashboard. 

IMPORTANT

We've put the last chapter of this Fast Track on the Mindware DVD. Please read it from there. Write to editor@thinkdigit.com and tell us whether you prefer reading on screen or on paper. The way in which you get Fast Tracks in the future depends on this, so make sure to write in!

7 CONTENT MANAGEMENT SYSTEM

If you've avoided using WordPress as a solution for building your own web site because you think it's only a blogging platform, and you don't want to have a blog (not every web site owner does, after all), it's time to rethink your position. WordPress is an able Content Management System (CMS) that is flexible and extensible enough to run an entire Web site — with no blog at all, if you prefer.

A CMS lets you maintain your web site with little or no knowledge of HTML. You can create, modify, retrieve, and update your content without ever having to touch the code required to perform those tasks. CMS programs such as WordPress give you the tools and advantages of blog software for managing and maintaining your site. These programs make setting up a web site much easier than in the past. Before, if you didn't know HTML, you had to hire a webmaster to maintain your web site. With WordPress, you can do all that yourself, with or without HTML knowledge. That capability translates into a huge cost savings and opens the door to anyone who wants to run a web site but doesn't have the resources or knowledge to do it.

This chapter shows you a few ways that you can use the self-hosted WordPress.org software to power your entire web site, with or without a blog. It covers different template configurations that you can use to create separate sections of your site. You also discover how to use the front page of your site as a landing page (a static page) or portal (a page that contains snippets from other sections of your site, with links to those sections), which can include a link to an internal blog page, if you want a blog.

7.1 Creating the static page

By default, a WordPress home page shows chronological blog post entries with the most recent post at the top. If your goal is to have a WP-created Page with static information, WordPress will allow you to select a different page as your home page so that you can display more traditional content like information about yourself or your business.

In your admin area, just go to **Setting > Reading**. Here you can decide if your home page will display your blog posts or a static page, if you choose a static page you can also choose which page to be your home page from the select box.

•First problem you may run into when you make a static page your home page is that the link to that page now appears in the main site navigation. Usually the page called “Home” in the main site navigation.



Reading Settings

Front page displays

☐ Your latest posts

☒ A static page (select below)

Front page: - Select -

Posts page: - Select -

About

Advertise

For a web site with impact, you need to have static pages as well

•The second problem is you will see that link again in the sidebar widget, the fix is pretty simple. The WordPress Pages navigation widget now allows you to exclude any page from its links by inserting the Page IDs of pages you want to exclude into the “Exclude” box in the Pages widget.

To have a static page display on the front page of your site, you need to create that page. Follow these steps:

1. Click the Add New link on the Pages menu.

The Add New Page page opens where you can write a new page to your WordPress blog.

2. In the Title text box, type a title for the page.

3. Type the content of your page in the text box.

4. Set the options for this page.

5. Click the Publish button.

The page is saved to your database and published to your WordPress site with its own, individual URL (or permalink). The URL for the static page consists of your blog URL and the title of the page. For example, if you titled your page “About Me,” then the URL of the page is `http://yourdomain.com/about-me`.

7.2 Assigning a static page as the front page

Next, you need to tell WordPress that you want the static page to serve as the front page of your site. Follow these steps:

1. Click the Reading link on the Settings menu to display the Reading Settings page.
2. In the Front Page Displays section, select the A Static Page radio button.
3. From the Front Page drop-down menu, choose the static page that you want to serve as your front page.
4. Click the Save Changes button at the bottom of the Reading Settings page. WordPress displays the page you selected in Step 3 as the front page of your site.

7.3 Tweaking the layout

With a little adjustment of the code in the `page.php` (Page) template, you can create a front page that doesn't look like a blog at all. However, you can use the quick-'n'-dirty method to change your front page from a bloggy-looking page to a nonbloggy-looking page using the WordPress Default Theme. Just follow these simple steps:

1. Click the Editor link on the Appearance menu.

The Edit Themes page opens. All the templates included in the current theme are listed on the right side of the page. By default, the Stylesheet (`style.css`) is displayed on the left side of the page in the text box.

2. Click Page Template link on the right side of the page.

The Edit Themes page refreshes, with the contents of the `page.php` template displayed in the text box on the left side of the page.

3. Locate this code:

```
<div id="content" class="narrowcolumn">
```

4. Replace the code in Step 3 with this code:

```
<div id="content" class="widecolumn">
```

The `widecolumn` class changes the formatting of the page content so that it is wider in width than the `narrowcolumn` class.

5. Remove the following code, located near the bottom of the `page.php` template:

```
<?php get_sidebar(); ?>
```

Removing this piece of code removes the sidebar from the static pages on your site. Without this code, the page displays only the content of your page.

6. Click the Update File button.

The file is saved, and your site is updated with the changes you just made.

If you want to add a link to your blog (see the next section), add the following bit of code to the content of the static page you created for your front-page display:

`Please Visit My Blog`

7.4 Adding a blog to your web site

If you want a blog on your site but don't want to display the blog on the front page, you can add one with the WordPress Administration panel. To create the blog for your site, follow these steps:

1. Click the Add New link on the Pages menu.

The page where you can write a new post to your WordPress blog opens.

2. Type Blog in the Title text box.

Doing this automatically sets the page slug to `/blog`.

3. Leave the Page Content box blank.

4. Click the Publish button.

The page is saved to your database and published to your WordPress site.

Now you have a blank page that redirects to `http://yourdomain.com/blog`.

Next, you need to assign the page you just created as your blog page. To do so, follow these steps:

1. Click the Reading link on the Settings menu to display the Reading Settings page.

2. From the Posts Page drop-down menu, choose the page you just created.

The page is set as your blog page.

3. In the Blog Pages Show at Most section, type the number of posts you want to displayed in the Posts text box.

This setting specifies the number of posts you want to display on that page at any time. If you enter 5, the blog page shows the last five posts you've made to your blog.

4. Click the Save Changes button.

The options you just set are saved, and your blog is now at **`http://yourdomain.com/blog`**.

This method of using the `/blog` page slug works only if you're using custom permalinks with your WordPress installation. If you're using the default permalinks, the URL for your blog page is different; it looks something like `http://yourdomain.com/?p=4` (where 4 is the ID of the page you created for your blog). You can find the URL for a page by clicking the Edit link on the Pages menu. Look for the page you created for your blog display, and click the View link that appears underneath the page title. The page opens in your browser, and you can copy the URL from the address bar in your browser and paste it into the code used in your static page to provide a link to your blog, like this:

```
<a href="http://yourdomain.com/blog">Please Visit My
Blog</a>.
```

7.5 Defining specific templates for static pages

A static page contains content that is not displayed on the blog page, but as a separate page within your site. You can have numerous static pages on your site, and each page can have a different design, which is governed by the template you create. You can create several static-page templates and assign them to specific pages within your site by adding code to the top of the static-page templates. Using a template on a static page is a two-step process: Upload the template, and tell WordPress to use the template by tweaking the page's code.

7.5.1 Uploading the template

To use a page template, you have to create one. You can create this file in a text-editor program, such as Notepad. To create an About page, for example, you can name your template `about.php`. When you have your template created, follow these steps to make it part of WordPress:

1. Upload the template file to your WordPress theme folder.

You can find that folder on your Web server in `/wp-content/themes/`.

2. Log in to your WordPress Administration panel, click the Editor link on the Appearance menu.
3. Click the `about.php` template link located on the right side of the page.
4. Type the Template Name tag directly above the header template tag.

The header tag should be similar to `<?php get_header(); ?>`.

If you're creating an About Page, the code to create the Template Name looks like this:

```
<?php
/*
Template Name: About Page
/*
?>
```

5. Click the Update File button. The file is saved and the page refreshes. If you created an About Page template, the `about.php` template is now called the About Page in the template list on the right side of the page. When you have the template created and named the way you want it, you assign that template to a page by following these steps:

1. Click the Add New link on the Pages menu.

The Add New Page page opens, where you can write a new post to your WordPress blog.

2. Type the title in the Title text box and the page content in the text box.
3. Choose the page template from the Page Template drop-down menu.

The Page Template drop-down menu in the Attributes module by default is shown on the right side of the page. You can reposition the modules on this page.

4. Click the Publish button to save and publish the page to your site.

7.5.2 Creating a template for each Post category

You don't have to limit yourself to creating a static-page template for your site. No, you can completely gorge yourself at the table of WordPress templates and create unique sections for your site.

You can create category templates for all categories in your blog simply by creating template files with filenames that correspond to the category ID numbers and then uploading those templates to your WordPress themes directory.

7.5.3 Pulling in content from a single category

WordPress makes it possible to pull in very specific types of content on your Web site through the `<query_posts>` template tag. If you have a category called WordPress, and you want to display the last three posts from that category on your front page, in your blog sidebar, or somewhere else on your site, you can use this template tag.

The `<query_posts>` template tag has several parameters that let you display different types of content, such as posts in specific categories and content from specific pages/posts or dates in your blog archives.

7.5.4 Customised Navigation bar

Adding too many pages to a WordPress blog that has a navigation bar in the header can really make your WordPress blog a mess. There are options to control what pages are shown in the navigation bar and even a way to add external links.

The Template Tag, `wp_list_pages()`, displays a list of WordPress Pages as links. It is often used to customize the Sidebar or Header. You can simply edit the `header.php` file and exclude any page id you want, you can also include the pages you want. So now the code of the top nav bar will look like this:

```
<ul id="pagenav">
    <li class="<?php if ( is_home() ) { echo 'current_
page_item'; } ?>">
        <a href="<?php bloginfo('url'); ?>" title="<?php
bloginfo('name'); ?>">Home</a>
    </li>
    <?php wp_list_pages('sort_column=menu_order&depth=1&title_
li=&exclude=17,38' ); ?>
</ul>
```

A common navigational scheme, parent pages on top and child pages (if they exist) on bottom:

You can 1) query the page, 2) determine if there are child pages, and 3) properly highlight both the `.current_page_parent` and `.current_page_item` links with this:

```
<ul id="nav">
<?php wp_list_pages('title_li=&depth=1'); ?>
</ul>

<?php if ($post->post_parent)
$children = wp_list_pages("title_li=&child_of=".$post-
>post_parent."&echo=0"); else
$children = wp_list_pages("title_li=&child_of=".$post-
>ID."&echo=0");
if ($children) { ?>
<ul id="subnav">
<?php echo $children; ?>
</ul>
<?php } else { ?>
<?php } ?>
```

You can even use CSS to make sure the `:active` and `:hover` states display correctly whether or not subpages exist — if they do, the primary nav uses `current_page_parent`, if they don't, it resorts to simply `current_page_item`.

7.5.5. Widgetise your theme

WordPress Widgets are also known as “sidebar accessories” which are WordPress Plugins or add-ons to your WordPress blog sidebar. This could come in handy for your client, WordPress Widgets allow the easy

addition of design elements, gadgets, content, images, and more to your WordPress sidebar to personalise the blog without knowing HTML, PHP, or any code.

Many themes have a widgetised sidebar or even two. But what about widgetising your WordPress footer. Sometimes it's a good to have widgets at the bottom of the blog for links to other posts and maybe some details about yourself. Let us see how to create 3 footer widget areas which can be edited from the admin area of your WordPress installation. Simply open your `functions.php` file in your current theme folder, and replace the current sidebar widget code with the one below:

```
<?php
if ( function_exists('register_sidebar') )
    register_sidebar(array(
        'before_widget' => '<li id="%1$s" class="widget
%2$s">',
        'after_widget' => '</li>',
        'before_title' => '<h2 class="widgettitle">',
        'after_title' => '</h2>',
    ));
    register_sidebar(array('name'=>'subfooterleft'));
    register_sidebar(array('name'=>'subfootercenter'));
    register_sidebar(array('name'=>'subfooterright'));
```

What this does is registers 3 sidebars called `subfooterleft`, `subfootercenter` and `subfooterright` which will correspond to the left, center and right widgets in the subfooter. You can then style the footer widgets, add the markup and finally add the widgets through the Admin area.

You can even widgetise your page menu in WordPress. Your theme needs to use the `wp_page_menu()` function to add its page menu so you can completely widgetise your menu.

Open your theme's `functions.php` file and add the code below:

```
<?php
register_sidebar( array(
    'name' => 'Page Menu',
    'id' => 'page-menu',
    'before_widget' => '<div id="page-nav">',
    'after_widget' => '</div>',
    'before_title' => false,
```

```
'after_title' => false
) );

add_filter( 'wp_page_menu', 'my_page_menu' );

function my_page_menu( $menu ) {
    dynamic_sidebar( 'page-menu' );
}
?>
```

Now, head over to the Widgets page in your WordPress admin. Select the Page Menu widget area and create your menu.

7.5.6 Custom post and page templates

Page templates let you change the design and/or functionality of particular pages by using a certain template. This new Page Template will then override the default page.php Page Template

included with your Theme. If you want to use another page template, simply open your

Write Page admin panel » Attributes and select the template you want for that page.

IMPORTANT

We've put the last chapter of this Fast Track on the Mindware DVD. Please read it from there. Write to editor@thinkdigit.com and tell us whether you prefer reading on screen or on paper. The way in which you get Fast Tracks in the future depends on this, so make sure to write in!

You can use a specific post template for single posts that you want to function differently. If you don't want the posts in a specific category to look or behave like the rest of the posts, you need to have another template for the posts in that specific category. You will need to put only the code below in your single.php file in your current theme folder to do that:

```
<?php
$post = $wp_query->post;
if ( in_category('13') ) {
    include(TEMPLATEPATH . '/single13.php');
} else {
    include(TEMPLATEPATH . '/single1.php');
```

```
}  
?>
```

Here we are telling WordPress: If the post is in category ID number 13, display `single13.php`. If not in category ID number 13, display `single1.php`. Now you just create 2 new files “`single13.php`” & “`single1.php`”, and do whatever you want to do. From this, you can make as many single post page looks as you want, as long as they are styled by their category.

7.5.7 Disable Comments and Trackbacks

If you want to disable comments on your blog, go to Discussion Settings » Default article settings to disable the comments on your blog. You can also edit your `single.php` file and delete the code that calls the `comments.php` file. **d**

8 DEALING WITH DATABASES

8.1 Understanding the WordPress database class

If you have developed Web sites or applications with PHP for any length of time, you are probably familiar with PHP database functions such as `mysql_query()`, `mysql_fetch_array()`, and so on. You might also be familiar with database abstraction classes. Like other abstraction classes, the database class used by WordPress provides an abstraction layer between the developer and the raw MySQL functions in PHP.

A database abstraction layer is a class or library that sits between your code and a series of databases. As every database has its own nuances, and the built-in database functions are messy and inconsistent, a database abstraction layer sorts out that messiness by providing a consistent interface to multiple databases.

The database class used by WordPress — ezSQL — is an extremely lightweight MySQL database abstraction class that, like much of WordPress' core, originated elsewhere. Before diving too far in, you should take note that ezSQL is not a database abstraction layer; it is a MySQL abstraction layer. It is possible, however, to take the `wp-db.php` file and manipulate it to talk to other database systems such as PostgreSQL or Microsoft's SQL Server. If you do this, simply save the file as `db.php` and drop it in the `wp-includes/` folder. WordPress looks to see if a `db.php` file exists first when bootstrapping, the process of getting WordPress connected to the database, and will use it instead of `wp-db.php` if it exists. You still need to know and understand MySQL or your underlying database. The class merely makes it easier to leverage.

ezSQL is not a full-featured abstraction engine like, for example, PHP 5's PHP Data Objects (PDO) library. It is a lightweight class that proves useful in many aspects of PHP development. It is perfectly suited to use within WordPress based on its licensing, size, and speed.

8.2 Examining the Schema and Properties of the database class

Because the database class is an object, it has quite a few properties and methods. Understanding these is critical for plugin developments because you will be using the database a lot. The database object is assigned to the handler `$wpdb` throughout WordPress. Whenever you create a function that requires transactions with the database, excluding the use of core

application programming interface (API) functions like `get_option()` or other similar functions, you need to ensure that you introduce `$wpdb` into the function or method using the global keyword.

Make the `$wpdb` object global whenever you write a function that needs to access the database

```
function my_function()  
{  
    global $wpdb;  
    // Function code here  
}
```

By making the `$wpdb` object available to your function, you expose all the properties and methods available to it.

The database properties

Every table and a variety of options are available as properties to be used with the database class. To access these properties, use `$wpdb->{property}` the same way you would use variables.

show_errors

The `show_errors` property is a true or false (also known as Boolean) property. By default it is set to false, but if you want to be able to debug SQL error messages in the process of development, set `show_errors` to true.

`$wpdb->show_errors` should always be set to false in production.

suppress_errors

The `suppress_errors` property is similar to `show_errors` in that it is a Boolean. In addition, it is similar because it is useful for debugging. However, this property pertains to suppressing database errors involved in bootstrapping.

last_error

The `last_error` property is a string that holds the last SQL or bootstrap error. It is useful for debugging, even if `$wpdb->show_errors` is false. By default, it is empty. This is not a log file and is reinitialized every time WordPress loads (page load).

num_queries

Like other properties already mentioned, the `num_queries` property is useful for debugging. It tracks the number of actual (not cached) queries that have been executed up to the moment it is called within WordPress. At

initialization, this is 0 but it is incremented with every real SQL query.

last_query

The `last_query` property, despite its name, does not store the SQL of the most recent query at the time of execution. Instead, it stores the last dataset of the last query that was run.

col_info

If you want to retrieve specific information about a column in a database table, use `$wpdb->col_info`. The `col_info` property stores information such as column type, name, and so on.

queries

Sometimes there is a need to save queries. It's not common in the WordPress world given WordPress provides its own caching mechanism; however, the capability exists. The `queries` property is an array of saved queries that requires the constant `SAVEDQUERIES` to be defined as true. Because `queries` is a private property, you'll never actually use it directly but the database class itself will.

prefix

The `prefix` property is commonly used in WordPress and is set to the table prefix that you defined in your `wp-config.php` file. By default, this prefix is `wp_`, but it can be anything.

Plugin authors should also take note that `prefix` is a very important property. When creating queries within your functions for custom tables, never assume that table prefixes are `wp_`. Always use `$wpdb->prefix` instead. This ensures that your plugin is compatible with all possible WordPress installs, including WordPress MU blogs.

ready

Another internal (private) property, `ready`, will never be used outside of the database class itself. This property is set to false by default, and is internally set to true when queries are ready to be executed.

posts

The `posts` property holds the name of the posts table, including the table prefix. Most often, it is set to `wp_posts`.

In a WordPress data context, it is important to note that several types of content are included in the post table.

Posts in a data context could include blog posts (default), pages, and attachments like image uploads, and so on. It is conceivable that plugin authors could provide their own new type of post. The type of post (post, page, revision, attachment) is determined by the `post_type` column. WordPress ignores `post_types` it doesn't know about, making this a good place to store data, as an alternative to creating your own tables.

postmeta

The `postmeta` property holds the name of the `postmeta` table, including the table prefix. Most often, it is set to `wp_postmeta`. The database schema for the `postmeta` table, which includes additional data related to individual posts. Generally, these are custom fields but plugin authors could use the table for storing their own metadata relating to posts.

users

The `users` property holds the name of the user table in WordPress, including the table prefix. Most often, it is set to `wp_users`. It is one of two tables that include all the pertinent information about the users in your WordPress

blog.

The `commentmeta` property contains the name of the table containing metadata about individual comments. It can be used by plugin authors to add ad hoc data about comments to the database. The table schema for the `commentmeta` table was added in WordPress 2.9 and is listed here in

options

The `options` table collects and stores all the settings for WordPress and many plugins. The name of this table, including the table prefix, is stored in the `options` property. Plugin authors should feel comfortable using this table for setting storage. It can be manipulated with `add_option()`, `update_option()`, and `get_option()`.

terms

The `terms` property stores the name, including table prefix, of the terms table. The terms table stores metadata that are referred to generically as terms. Terms can be categories, tags, or link categories and can be extended to other areas by plugin developers

tables

The `tables` property is private and, thus, shouldn't be accessed directly, if at all. The property contains an array of the table names that are used by WordPress.

field_types

The `field_types` property is new in WordPress 2.8. It contains an array of field names where you can specify what character type (integer or string) to return field data as. This is set in `wpsettings.php`, but can be re-declared in a plugin function that hooks into the `init` action hook.

charset

Like the table properties, the `charset` property stores the character set of the database itself, which is generally UTF-8. The `charset` property is unlikely to be used often in WordPress plugin development.

collate

Similar to the `charset` property, the `collate` property holds the collation value for the database. It is unlikely to be used often in WordPress plugin development.

real_escape

The final property available in the database class, the `real_escape` property, instructs WordPress on whether to use the `mysql_real_escape_string()` function. The function is set to `false` by default because data sanitization is handled elsewhere in WordPress, making the need for this property unlikely.

8.3 Adding Data to MySQL with WordPress

Two main API methods are used to add data to MySQL: `insert()` and `update()`. These methods take care of sanitizing and preparing all data for insertion or updating MySQL from within WordPress. For the sake of backward compatibility, WordPress waits a long time before removing functions, classes, methods and, in this case, table name properties from the core. This is because plugin authors, in many cases, try to build plugins that will work on any version that has been released. When WordPress 2.3 was released, it introduced the concept of tags when before, the only metadata available for posts was in the form of categories. During those days, there were a number

of tables that existed, but were replaced with the “terms” tables instead. categories

The categories table existed to hold category names, IDs, and descriptions. Even though categories still exist, they are now stored as terms. The database class maintains the `$wpdb->categories` property to hold the name of this deprecated table.

post2cat

The `post2cat` table existed as a join table to associate posts with the categories they were assigned to. This table was eliminated with the advent of the terms infrastructure. The database class maintains `$wpdb->post2cat` to hold the name of this deprecated table.

Inserting new data into MySQL

To insert data into MySQL, you can safely use the `insert()` method. This method takes three arguments: the table name, an array of column/value pairs, and an optional format array (%s or %d) that will force the data to be either a string or number.

The data that is going to be inserted does not need to be sanitized, and in fact, should be “raw” — that is, unescaped. WordPress will handle all the heavy lifting on this.

Using the `insert()` method to insert a post with a possible custom post type, mp3

```
function add_music_post( $music_title, $music_description
)
{
global $wpdb;
$wpdb->insert( $wpdb->posts, array('post_title' => $music_
title, 'post_content' => $music_description, 'post_type'
=> 'mp3'), array('%s', '%s',
'%s') );
}
```

8.4 Updating data in MySQL safely

A second data insertion method is the `update()` method. This method, like the `insert()` method, is intended to make the process of changing data in MySQL a safe process free of SQL injection. It carries a similar format as the `insert()` method but adds a few additional arguments.

The first argument is the table name, the second argument is an array

of data to be changed in field/value pairs, and the third argument is an array of field/value pairs used to provide “where” conditions for the update. All three of these arguments are required.

The first two arguments are identical to the insert() method. The fourth and fifth arguments, respectively, behave similarly to the format argument in insert(). The fourth argument is an array of data formats (%s for string and %d for number) that forces the data that will be inserted into the database to be treated as the appropriate data type. The fifth argument is identical, except it pertains to the data used in the WHERE clause.

Using the update() method to safely update data in MySQL

```
function edit_music_post( $post_id, $new_title, $new_
description )
{
global $wpdb;
$wpdb->update( $wpdb->posts, array( 'post_title' => $new_
title, 'post_
content' => $new_content ), array( 'ID' => $post_id ),
array( '%s',
'%s'), array( '%d' ) );
}
```

8.5 Retrieving data from MySQL with WordPress

It's great that the database class has so many properties but what good is a Web application if it can't read and write to a database? Of course, WordPress can and does. A number of methods exist that will retrieve data in an elegant, object-oriented way if necessary. It is always good practice to avoid having to include raw SQL in a plugin when possible. The reality is that plugins that rely on raw SQL are often the ones that break later on. A good example of this is the myriad of plugins that used the categories table prior to WordPress 2.3. When the \$wpdb->categories and \$wpdb->post2cat tables disappeared and were replaced by the new terms paradigm, the plugins that were querying the categories tables directly broke. In many cases, WordPress provides methods and functions to retrieve data from MySQL as part of its plugin API. Where this API exists, you should use it because it is generally forward compatible.

Retrieving a single value from a table

In the event that an API doesn't exist to retrieve the data needed, you can use

the `$wpdb->get_var()` method to retrieve a single value from the database. Simply pass your SQL statement to the method. The SQL that is passed must return only a single field from a single row like this:

```
$post_id = $wpdb->get_var("SELECT ID FROM ` ` . $wpdb->posts  
    . ` ` WHERE  
post_author = 1 LIMIT 1");
```

In plain English, this query will retrieve the ID of the most recent post written by the author with an ID of 1. You'll notice that this line of code uses the `$wpdb->posts` property as well. If you recall, this property held the name of the posts table. You'll also notice that the query itself is only retrieving a single field (ID) from a single row (in this case, you added `LIMIT 1` to the query).

Retrieving a column of data

What if you wanted to get all the IDs of the posts by the author with an ID of 1? For this use case, you want to use the `$wpdb->get_col()` method. When only one field will be retrieved, but there will be more than one row of data, you need to use this method.

By default, the data that is returned will be in the form an object; however, you can pass the method a second argument of `ARRAY_A` or `ARRAY_N` to have data returned as an associative or numerical array.

Using the `get_col()` method

```
$object = $wpdb->get_col("SELECT ID FROM ` ` . $wpdb->posts  
    . ` ` WHERE post_  
author = 1");  
$assoc_array = $wpdb->get_col("SELECT ID FROM ` ` . $wpdb->  
posts . ` ` WHERE  
post_author = 1", ARRAY_A);  
$num_array = $wpdb->get_col("SELECT ID FROM ` ` . $wpdb->  
posts . ` ` WHERE post_  
author = 1", ARRAY_N);
```

Retrieving a row of data

Similar to `$wpdb->get_col()`, the database class also provides `$wpdb->get_row()`, a method for retrieving only a single row of data. Ignoring the fact that WordPress provides a `get_userdata()` function as an API function to retrieve information about a user, you might use this method to retrieve information about a user with ID 43.

Using the `get_row()` method

```
$userdata = $wpdb->get_row("SELECT * FROM ` ` . $wpdb->users
. ` WHERE ID =43");
$user_assocarray = $wpdb->get_row("SELECT * FROM ` ` .
$wpdb->users . ` WHERE ID = 43", ARRAY_A);
$user_numarray = $wpdb->get_row("SELECT * FROM ` ` . $wpdb->
>users . ` WHERE ID= 43", ARRAY_N);
```

Retrieving a full data set

If you have to do raw SQL queries, try to limit the scope of what data you are actually retrieving from the database. If you can just retrieve a column of data, then use `$wpdb->get_col()`. If you only need a single value, consider the `$wpdb->get_var()` method.

However, sometimes you will need an entire dataset. Fortunately, you have the `$wpdb->get_results()` method. Like the `$wpdb->get_col()` and `$wpdb->get_row()` methods, you can pass `ARRAY_A` or `ARRAY_N` as a flag so the returned data comes back as an array.

Using the `get_results()` method

```
$users = $wpdb->get_results("SELECT * FROM ` ` . $wpdb->
>users . ` WHERE user_registered > '2009-07-01'");
$users_assocarr = $wpdb->get_results("SELECT * FROM ` ` .
$wpdb->users . `WHERE user_registered > '2009-07-01'",
ARRAY_A);
$users_numarr = $wpdb->get_results("SELECT * FROM ` ` .
$wpdb->users . ` WHERE user_registered > '2009-07-01'",
ARRAY_N);
```

Performing other queries

The WordPress database class provides a `$wpdb->query()` method for other types of SQL queries. For example, you can use it to get a listing of all tables in the database (something that might be helpful because other plugins might supply their own tables):

```
$wpdb->query("INSERT INTO {$wpdb->prefix}my_plugin_table SET field1
=
'$safe_value', field2 = '$safe_value2' WHERE field3 = 'some
value'");
```

You could also use this method for INSERT, DELETE, or UPDATE SQL queries if your plugin provides its own tables.

8.5 Preventing SQL injection

If you've spent any serious time developing Web applications, you know of the risk of SQL injection. SQL injection is caused when data is inserted into the database that hasn't been properly sanitized. For example, a classic demonstration of SQL injection would be an exploit that gives a malicious person administrator access to your site.

It is hard to emphasize the importance of preventing SQL injection with your plugin. WordPress is only as secure as its weakest link. It provides many options to access data that do not require any SQL at all. If you can, it is imperative that you use these functions and API methods. You should only use SQL when it's absolutely essential. Using it without proper care can destroy someone's blog. I cannot emphasize this enough.

Say you have an HTML form that requires a username and password. Each of these form fields are named "user" and "pass," respectively. On the backend, the authentication mechanism checks the username and password against the database like this:

```
SELECT * FROM example_usertable WHERE username = '$input_user' AND password = '$input_pass';
```

If form data is unsanitized but the user actually inputs a proper username and password, then the SQL query works and authentication is granted. However, if someone enters a more malicious string, they can wreak havoc on your system. For example, if the user entered a username of `user' OR 1=1;--` and any password, then the SQL will also be valid and will give the user access to the protected system. The query that would actually be run is shown here:

```
SELECT * FROM example_usertable WHERE username = 'user' OR 1=1;--random password
```

To understand the practical danger in this query, you have to understand that two dashes (--) is equivalent to a comment. In essence, everything after the -- is ignored by MySQL so it doesn't matter if an invalid password is provided. It could be anything. Also, because you haven't "escaped" the single quote in the username, the single quote becomes the "end" of the value being checked against the username field. Though this is probably invalid as well, it doesn't matter because the attack adds an `OR 1=1` (which is always true) into the query. The query reads, "Retrieve the record from the usertable where the username is 'user' or where `1=1`." Because `1` always equals `1`, the SQL always evaluates as true and an attacker can access the protected resources, application or, possibly, a WordPress plugin.

IMPORTANT

We've put the last chapter of this Fast Track on the Mindware DVD. Please read it from there. Write to editor@thinkdigit.com and tell us whether you prefer reading on screen or on paper. The way in which you get Fast Tracks in the future depends on this, so make sure to write in!

To prevent against a possible SQL injection attack (which could be as simple, or far more complex, than this example), the WordPress database class provides the `$wpdb->prepare()` method. This method, added in WordPress 2.3, uses the PHP `printf()` syntax (<http://php.net/printf>) for field replacement. It handles data sanitization of SQL statements to prevent against SQL injection. Note that, like `printf()`, placeholders can use an “order” index.

Instead of using `%d` to indicate a numerical replacement, you could use the `%1$d` syntax to indicate, “replace this placeholder with the first replacement value in the format of a number.”

The two placeholders the `prepare()` method supports are `%d` for integers and `%s` for strings. Note that in neither case do you quote the placeholder. WordPress will take care of proper quoting for you. **d**